Hyperdimensional Computing

HDC

Content:

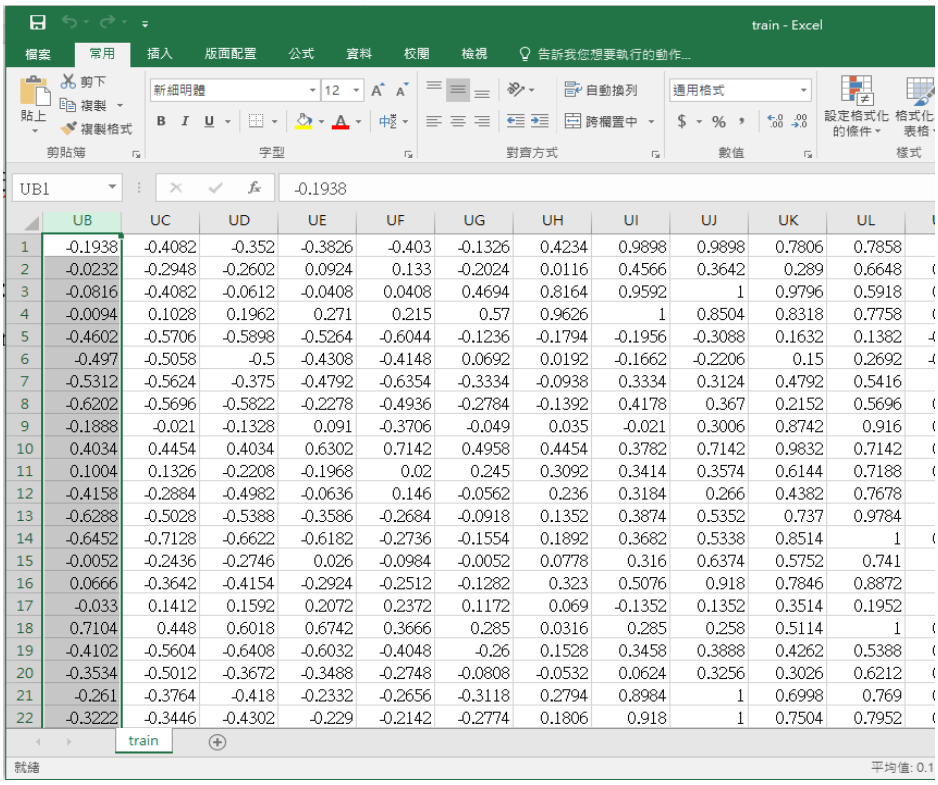
The accuracy with different dimension and CIM levels

B06901053 電機三謝承延

Dataset Introduction: ISOLET

ISOLET (Isolated Letter Speech Recognition) is a database of alphabets in English spoken separately. It contains 26 letters from a, b, c...to z. In this experiment, I will use HDC model with different dimension and CIM levels to inspect the accuracy of prediction results.

The csv file used for training consists of 6238 samples and 617 features, while the csv file used for testing consists of 1559 samples and 617 features.



	UB	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL
1	-0.1938	-0.4082	-0.352	-0.3826	-0.403	-0.1326	0.4234	0.9898	0.9898	0.7806	0.7858
2	-0.0232	-0.2948	-0.2602	0.0924	0.133	-0.2024	0.0116	0.4566	0.3642	0.289	0.6648
3	-0.0816	-0.4082	-0.0612	-0.0408	0.0408	0.4694	0.8164	0.9592	1	0.9796	0.5918
4	-0.0094	0.1028	0.1962	0.271	0.215	0.57	0.9626	1	0.8504	0.8318	0.7758
5	-0.4602	-0.5706	-0.5898	-0.5264	-0.6044	-0.1236	-0.1794	-0.1956	-0.3088	0.1632	0.1382
6	-0.497	-0.5058	-0.5	-0.4308	-0.4148	0.0692	0.0192	-0.1662	-0.2206	0.15	0.2692
7	-0.5312	-0.5624	-0.375	-0.4792	-0.6354	-0.3334	-0.0938	0.3334	0.3124	0.4792	0.5416
8	-0.6202	-0.5696	-0.5822	-0.2278	-0.4936	-0.2784	-0.1392	0.4178	0.367	0.2152	0.5696
9	-0.1888	-0.021	-0.1328	0.091	-0.3706	-0.049	0.035	-0.021	0.3006	0.8742	0.916
10	0.4034	0.4454	0.4034	0.6302	0.7142	0.4958	0.4454	0.3782	0.7142	0.9832	0.7142
11	0.1004	0.1326	-0.2208	-0.1968	0.02	0.245	0.3092	0.3414	0.3574	0.6144	0.7188
12	-0.4158	-0.2884	-0.4982	-0.0636	0.146	-0.0562	0.236	0.3184	0.266	0.4382	0.7678
13	-0.6288	-0.5028	-0.5388	-0.3586	-0.2684	-0.0918	0.1352	0.3874	0.5352	0.737	0.9784
14	-0.6452	-0.7128	-0.6622	-0.6182	-0.2736	-0.1554	0.1892	0.3682	0.5338	0.8514	1
15	-0.0052	-0.2436	-0.2746	0.026	-0.0984	-0.0052	0.0778	0.316	0.6374	0.5752	0.741
16	0.0666	-0.3642	-0.4154	-0.2924	-0.2512	-0.1282	0.323	0.5076	0.918	0.7846	0.8872
17	-0.033	0.1412	0.1592	0.2072	0.2372	0.1172	0.069	-0.1352	0.1352	0.3514	0.1952
18	0.7104	0.448	0.6018	0.6742	0.3666	0.285	0.0316	0.285	0.258	0.5114	1
19	-0.4102	-0.5604	-0.6408	-0.6032	-0.4048	-0.26	0.1528	0.3458	0.3888	0.4262	0.5388
20	-0.3534	-0.5012	-0.3672	-0.3488	-0.2748	-0.0808	-0.0532	0.0624	0.3256	0.3026	0.6212
21	-0.261	-0.3764	-0.418	-0.2332	-0.2656	-0.3118	0.2794	0.8984	1	0.6998	0.769
22	-0.3222	-0.3446	-0.4302	-0.229	-0.2142	-0.2774	0.1806	0.918	1	0.7504	0.7952

Experiment 1:

The accuracy between different dimensions

HDC uses hypervector as basic element to compute. This experiment focuses on the relationship between accuracy and dimension of hypervectors.

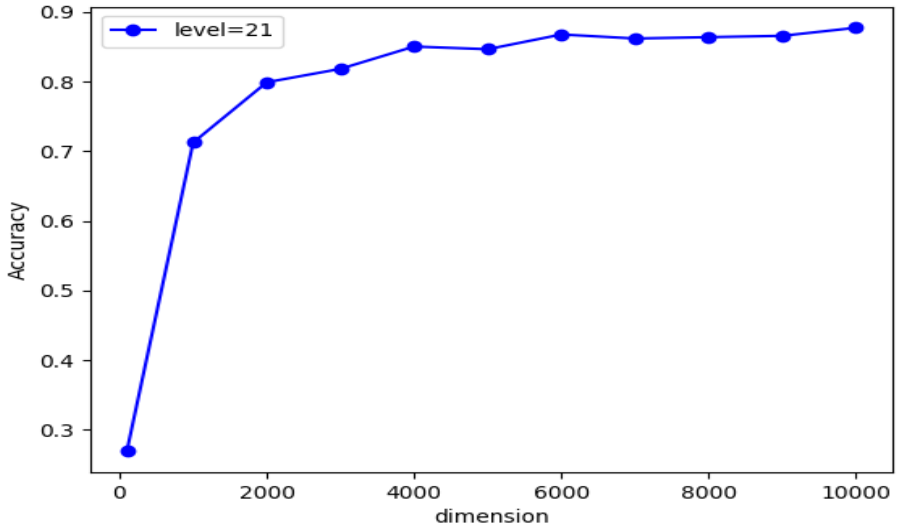
I choose 11 different dimensions in this experiment---[100, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000], respectively.

Also, I test these different dimensions with 20 different CIM levels (from 2 to 21) as well. That is, in each CIM levels, you can observe the relationship between accuracy and dimensions.

However, for the sake of simplicity, I will only introduce the results of 3 different levels in this report.

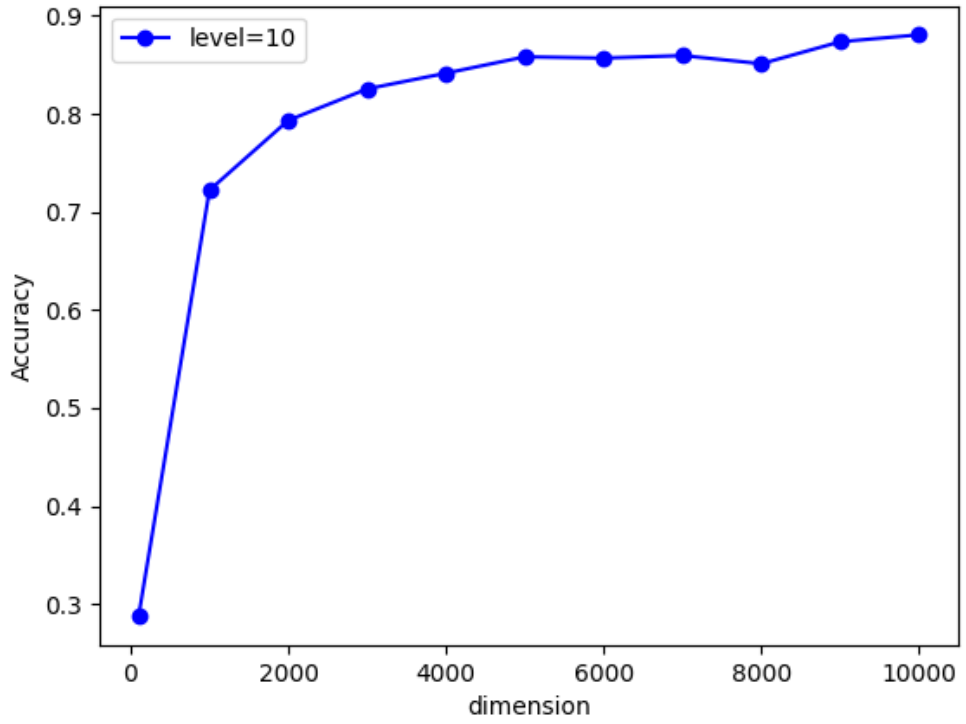
Results:

Level 21:



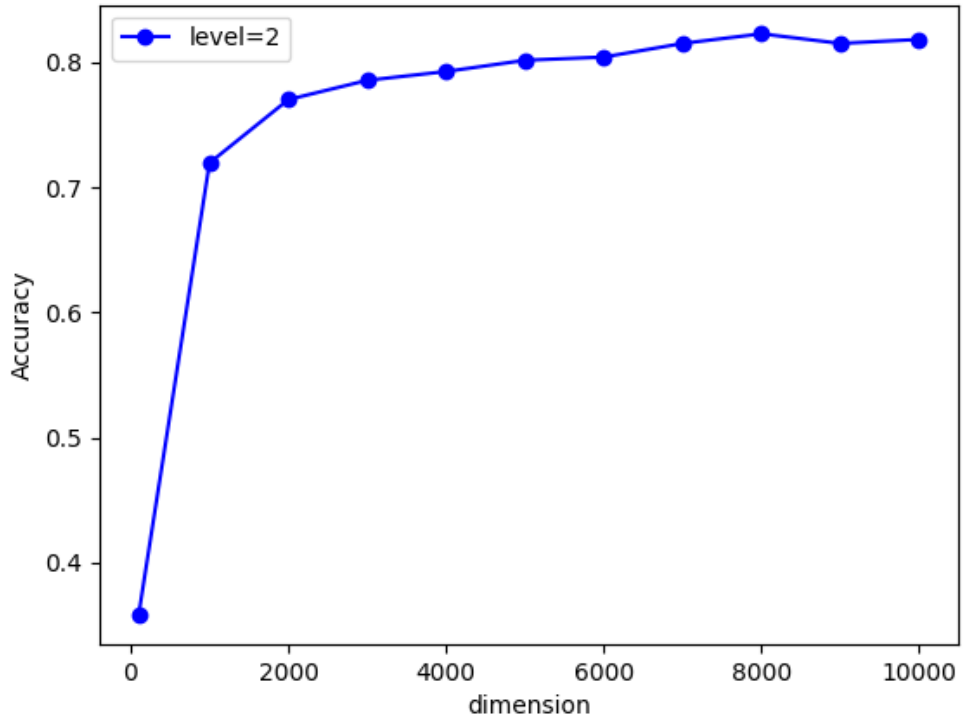
Dimension	Accuracy
100	0.269403
1000	0.712636
2000	0.79923
3000	0.818473
4000	0.850545
5000	0.846697
6000	0.867864
7000	0.862091
8000	0.864015
9000	0.86594
10000	0.877486

Level 10



Dimension	Accuracy
100	0.287364
1000	0.722258
2000	0.793457
3000	0.825529
4000	0.841565
5000	0.858242
6000	0.85696
7000	0.859525
8000	0.851187
9000	0.873637
10000	0.880693

Level 2:



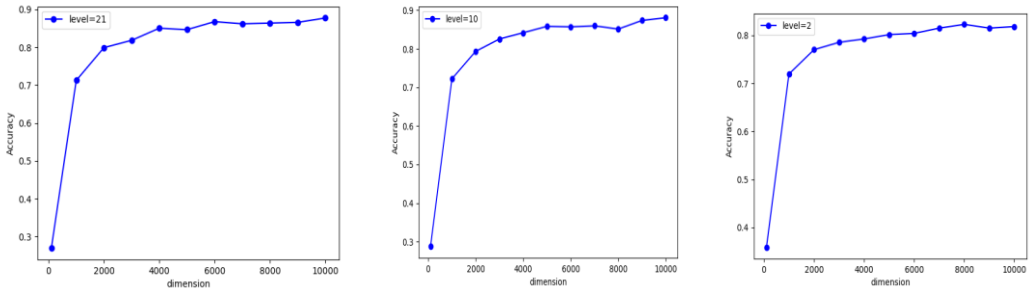
Dimension	Accuracy
100	0.357922
1000	0.719692
2000	0.770366
3000	0.78576
4000	0.792816
5000	0.801796
6000	0.804362
7000	0.815266
8000	0.822963
9000	0.815266
10000	0.818473

Conclusion:

Why presenting different CIM levels?

The above are the relationship between different dimension in three different CIM levels.

The reason why I present the results with additional CIM levels is to show you that the results from different dimension only slightly differed in different CIM levels.



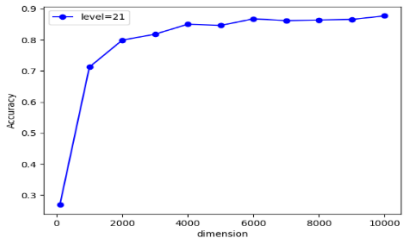
You can see that the variation trend between these three different CIM levels seem to be quite similar. As a result, we can conclude that choosing different CIM levels does not influence the results of this experiment.

So, lets focus on the relationship between dimension and accuracy.

The relationship between accuracy and dimension.

As you can see, the accuracy is the worst when we choose the smallest dimension 100. When the dimension changes from 100 to 1000, the accuracy rises rapidly from about 30% to 72%. The accuracy keeps rising with considerable speed when dimension increases until dimension 5000.

After entering dimension 5000, which achieves about 85% accuracy, the rise of accuracy with increase of dimension becomes smooth.



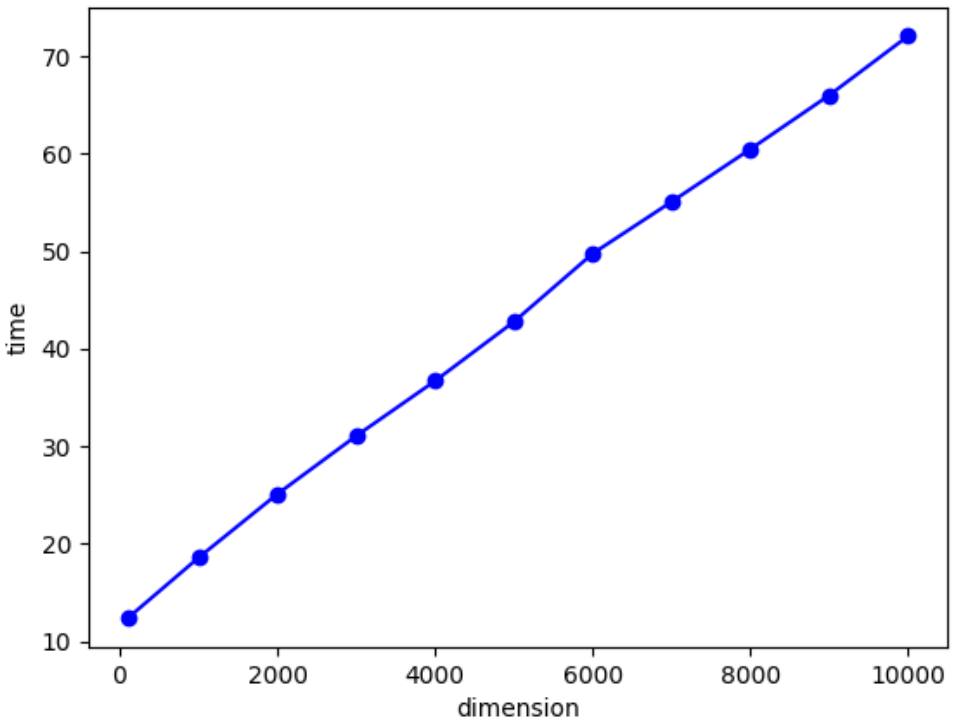
In most cases, Dimension 10000 achieves the best accuracy (about 88% with CIM level 21,15)。

From the observation, we can learn that HDC can rapidly achieve a certain degree of accuracy with only small amount of increase in the dimension. However, when increasing dimension to a certain level, the growth of accuracy becomes nonobvious.

The total training time with different dimension.

HDC is a very efficient computing method. It is known as its high speed and high efficiency. Comparing to NN, HDC can achieve comparable accuracy with much less time consumed.

In the last session, we learn that only increasing Dimension to 5000 is enough to achieve acceptable performance. In this session, we will focus on the training time between different dimension to observe the high efficiency of HDC.



Dimension	Time
100	12.37374
1000	18.59008
2000	25.11531
3000	31.07147
4000	36.70869
5000	42.77547
6000	49.76842
7000	55.07831
8000	60.47359
9000	65.99848
10000	72.06984

You can observe that training HDC with dimension 100 only takes 12 seconds, which is an astonishing results comparing with state-of-art ML method.

Training HDC with dimension 5000 takes 43 seconds approximately and can achieve 85% accuracy, while dimension 10000 takes 72 second to train and can achieve accuracy 88%.

From the observation, we can find that the advantage of HDC is that it can achieve great performance with only small amount of time. However, when increasing dimension to

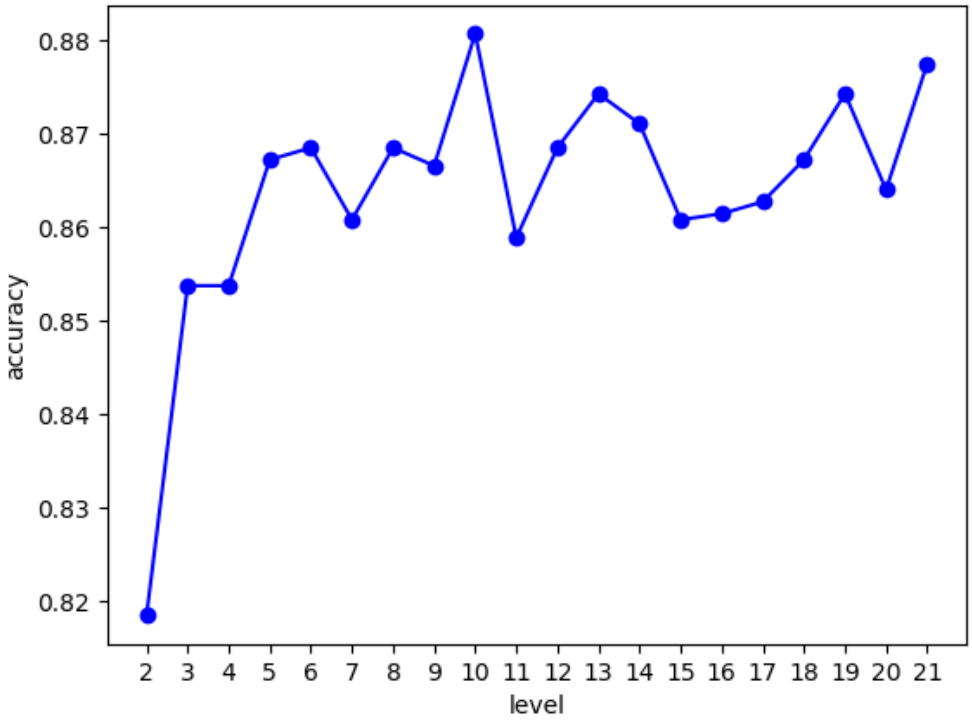
a certain level, the performance improvement slows down and the cost of time consumption keeps rising.

We can conclude that HDC is suitable for a case which requires acceptable performance with only small time cost.

Experiment 2:

The accuracy between different levels:

To use different levels in HDC model implies the number of section to partition the continuous value into. In this experiment, I choose dimension 10000 to test the impact of choosing different levels on the accuracy.



By instinct, when we partition the continuous value into more sections, which means larger levels, we can obtain better results since the information are distinguished more thoroughly.

Level	Accuracy
2	0.818473
3	0.853752
4	0.853752
5	0.867223
6	0.868505
7	0.860808
8	0.868505
9	0.866581
10	0.880693
11	0.858884
12	0.868505
13	0.874278
14	0.871071
15	0.860808
16	0.86145
17	0.862733
18	0.867223
19	0.874278
20	0.864015
21	0.877486

From the observation, we can find that with the increase of levels, the accuracy indeed increases. However, we can also find that the highest accuracy is not achieved when choosing the largest level 21, but level 10.

I think that one reasons behind this is about the orthogonality of CIM vector. In HDC, we choose the lowest value in each feature as an random hypervector, and we change $\frac{\#dimension}{2\ x\ (\#level-1)}$ elements each time until the highest value corresponds to a hyper vector orthogonal to the vector of the lowest value. However, when $\#dimension$ can not be divided by $2\ x\ (\#level-1)$, the vector of highest and lowest value will not be orthogonal.

The second reason I guess which may result in such condition is that partitioning the levels too thoroughly may sometimes assign similar properties into different level. For example, considering level=20 with value 1,2,20, you can see that 1 and 2 may be very similar properties. However, when level=20, value 1 and value 2 will be

assigned to two different vector. Such condition may results in the cases that similar things are classified as different since they are assigned to different vectors.