DHRUV GUPTA

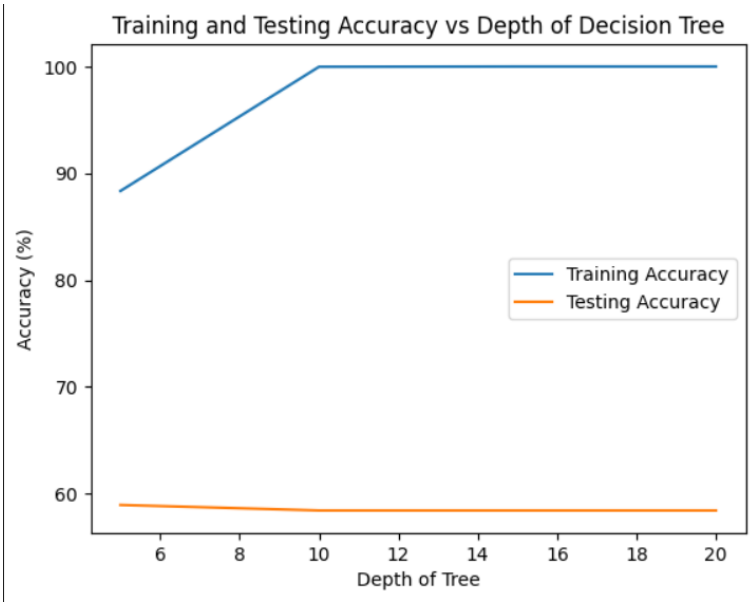
2023EE30858

COL 774

ASSIGNMENET – 3 (Decision Tree) Report

(A)

```
Train -accuracy for depth 5 :  88.34802606362591 %
Validation-accuracy for depth 5 :  56.896551724137936 %
Test-accuracy for depth 5 :  58.94519131334023 %
---------------------------------
Train -accuracy for depth 10 :  99.97444742557813 %
Validation-accuracy for depth 10 :  56.43678160919541 %
Test-accuracy for depth 10 :  58.42812823164426 %
---------------------------------
Train -accuracy for depth 15 :  100.0 %
Validation-accuracy for depth 15 :  56.32183908045977 %
Test-accuracy for depth 15 :  58.42812823164426 %
---------------------------------
Train -accuracy for depth 20 :  100.0 %
Validation-accuracy for depth 20 :  56.32183908045977 %
Test-accuracy for depth 20 :  58.42812823164426 %
---------------------------------
```

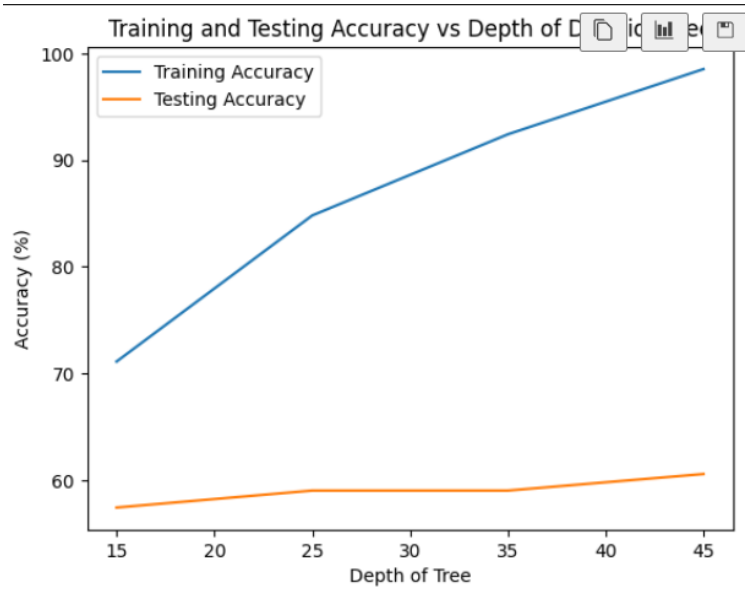


Comments -

We see that the decision tree tends to over fit as we increase the maximum possible depth it can attain. This makes it prone to the noise in the training data and then thus learns patterns which are not present there. It does not generalise well over all datasets, and thus we see a dip in the testing data accuracy on increasing the depth.

(B)

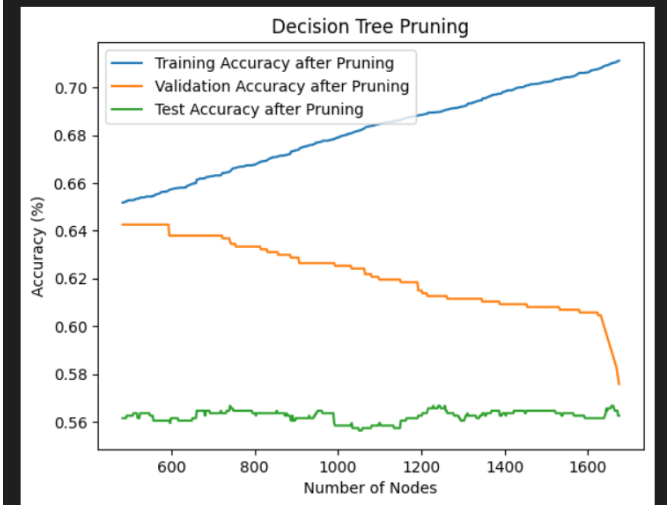
```
Train accuracy for depth 15: 71.13%
Validation accuracy for depth 15: 57.36%
Test accuracy for depth 15: 56.46%
Train accuracy for depth 25: 84.81%
Validation accuracy for depth 25: 60.11%
Test accuracy for depth 25: 59.05%
Train accuracy for depth 35: 92.41%
Validation accuracy for depth 35: 58.28%
Test accuracy for depth 35: 59.05%
Train accuracy for depth 45: 98.52%
Validation accuracy for depth 45: 58.97%
Test accuracy for depth 45: 60.60%
```



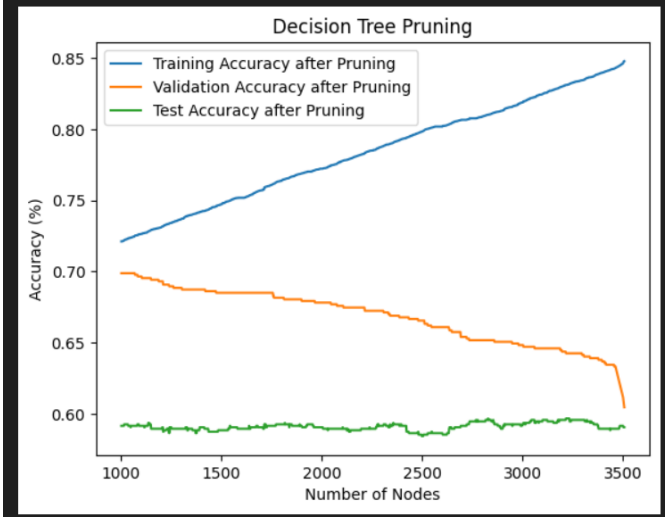
Comments - The data underfits for lesser depths i.e. it cannot capture all the patterns present in the data. Hence, on increasing the depth, similar to adding more parameters that the tree learns, in some sense, we increase the accuracy of the model because the now the tree has more complexity and can represent the decision boundaries for the data more accurately. In one-hot encoding, for the same depth the tree does not overfit, as in one-hot encoding a k-valued category is split into k binary classified variables and the binary classification requires more splits to achieve the k-way portioning. Meaning, that if I have 5 teams then the k-way split can split into 5 nodes, but the numerical classificationwill need more than 1 steps to achieve this. Thus the data compressed in one node/at one level is reduced and this means that the One Hot Encoded trees need more depth to overfit.

(C)

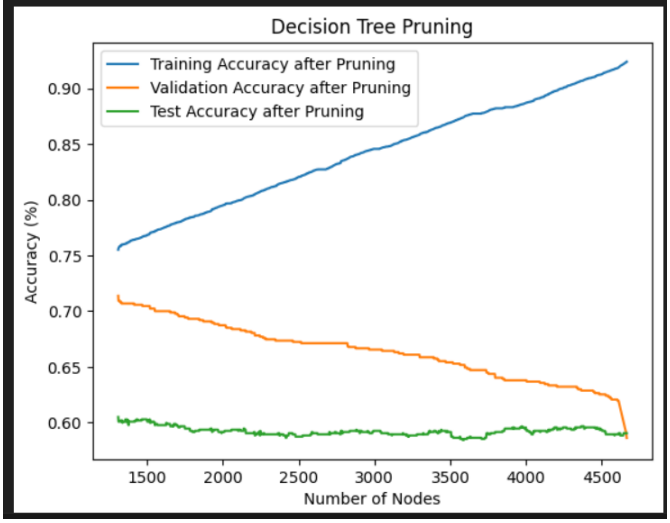
```
Training Accuracy after pruning for depth 15 :  0.651718410629871 %
Validation Accuracy after pruning for depth 15 :  0.6425287356321839 %
Test Accuracy after pruning for depth 15 :  0.56153050672182 %
```



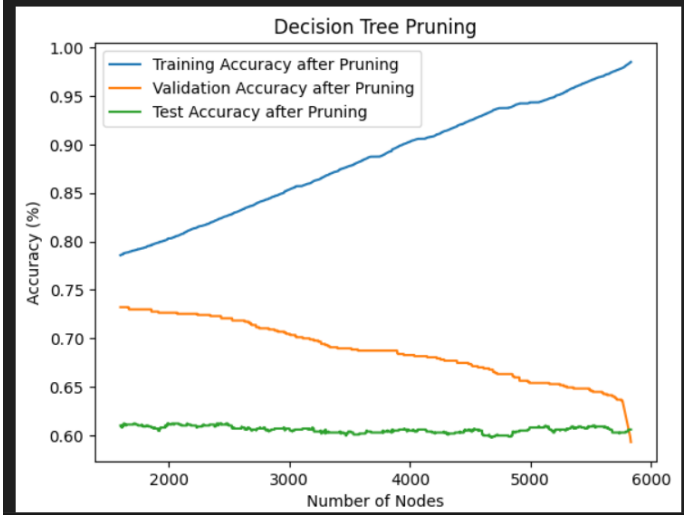

```
Training Accuracy after pruning for depth 25 :  0.7212214130573655 %
Validation Accuracy after pruning for depth 25 :  0.6988505747126437 %
Test Accuracy after pruning for depth 25 :  0.5915201654601862 %
```




```
Training Accuracy after pruning for depth 35 :  0.7552063370384566 %
Validation Accuracy after pruning for depth 35 :  0.7137931034482758 %
Test Accuracy after pruning for depth 35 :  0.6049638055842813 %
```



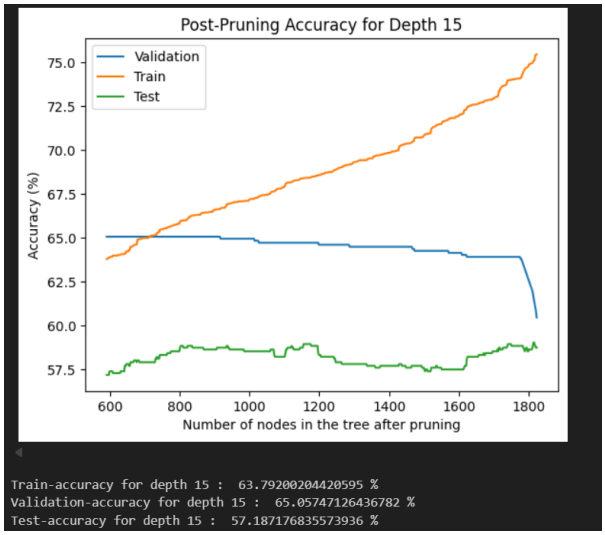

```
Training Accuracy after pruning for depth 45 :  0.7857416634725949 %
Validation Accuracy after pruning for depth 45 :  0.732183908045977 %
Test Accuracy after pruning for depth 45 :  0.610134436401241 %
```

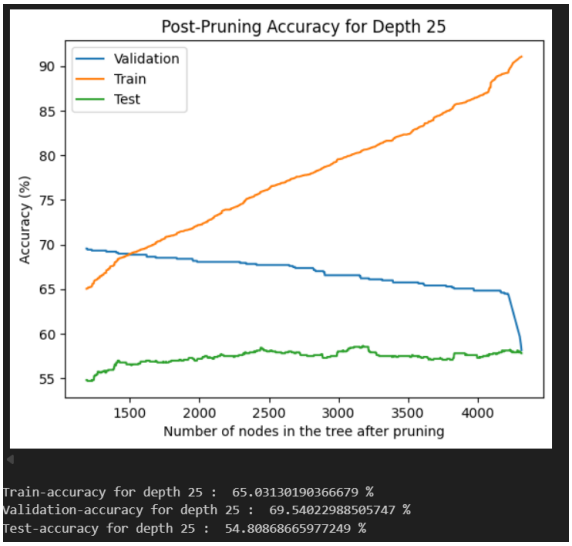


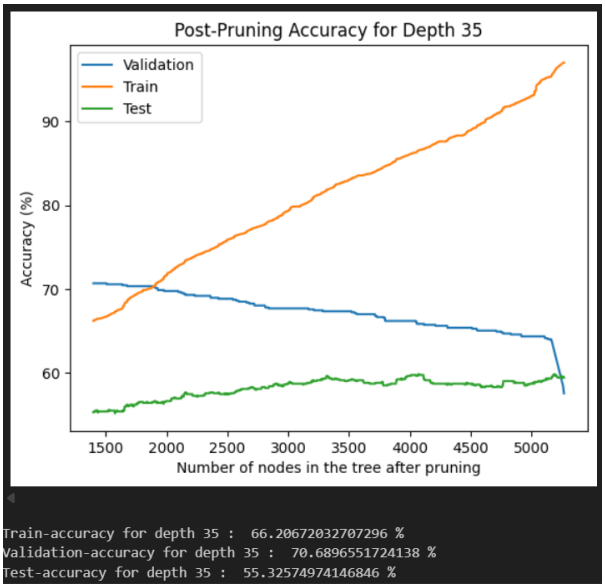
Comments - We see that the there is significant dip in the training data accuracies. This is because pruning removes the overfitting branches from the tree and allows the model to generalize better. We also see that deeper trees result in higher testing accuracies suggesting that deeper trees can better capture the underlying patterns present in the data. As a result, the model demonstrated better generalization on the validation dataset. However, this

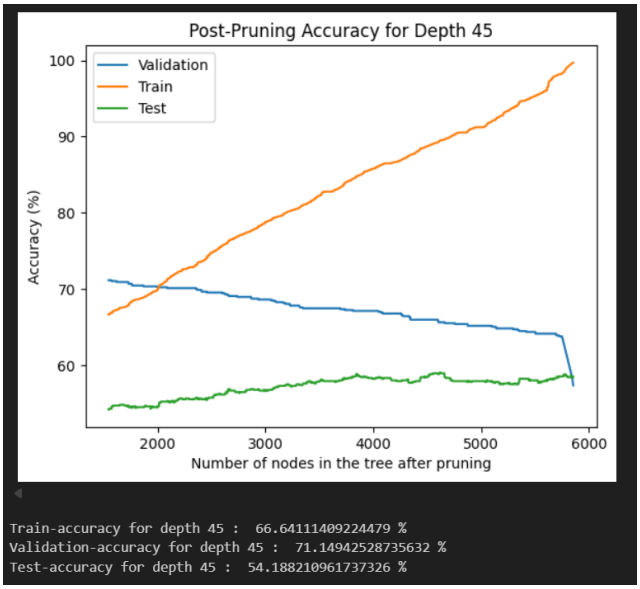
refinement led to a noticeable decrease in training accuracy and a slight reduction in test accuracy, reflecting the expected bias-variance trade-off introduced by pruning.

(D)



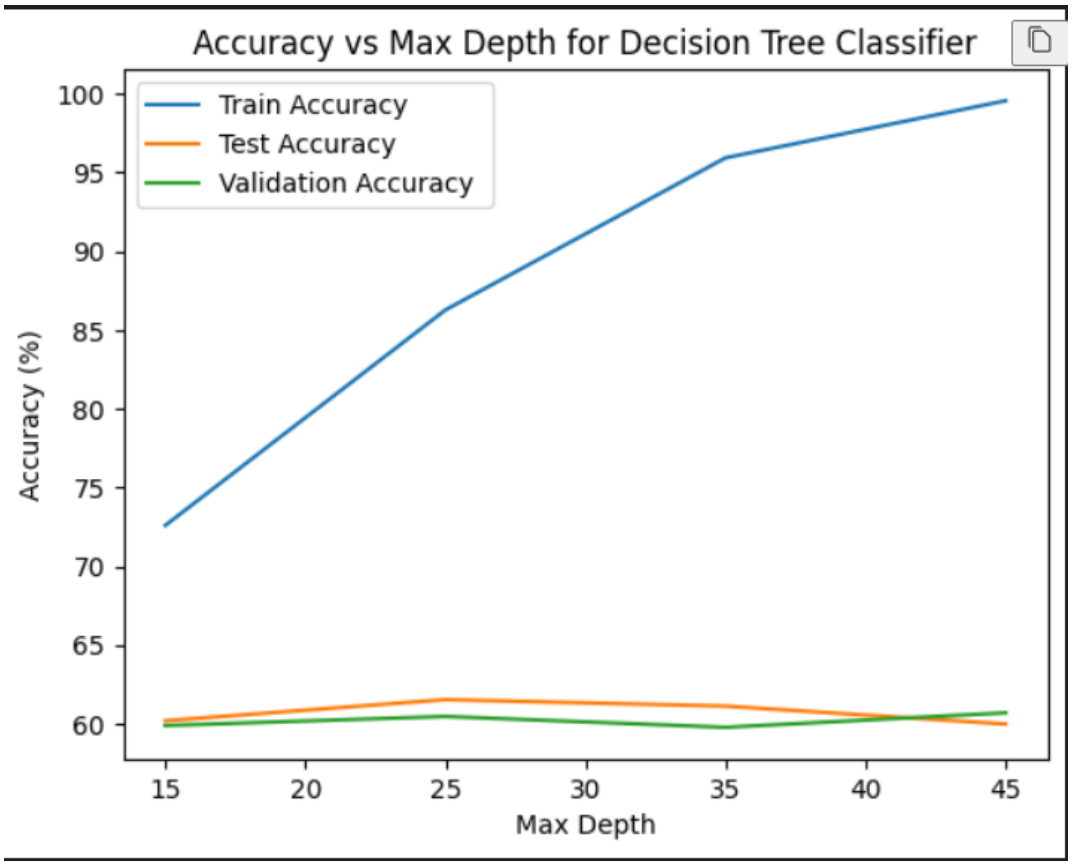






Comments - We see slightly higher accuracy as the gini criteria is results in better leaf nodes as it isolates the major class better. It is computationally faster than the entropy function. The test accuracies however take a sligh dip with reduction in nodes, majorly because we kind of try to overfir the data towards validation set, which surely does not capture entire distrbution due to lack of data when compared to training set.

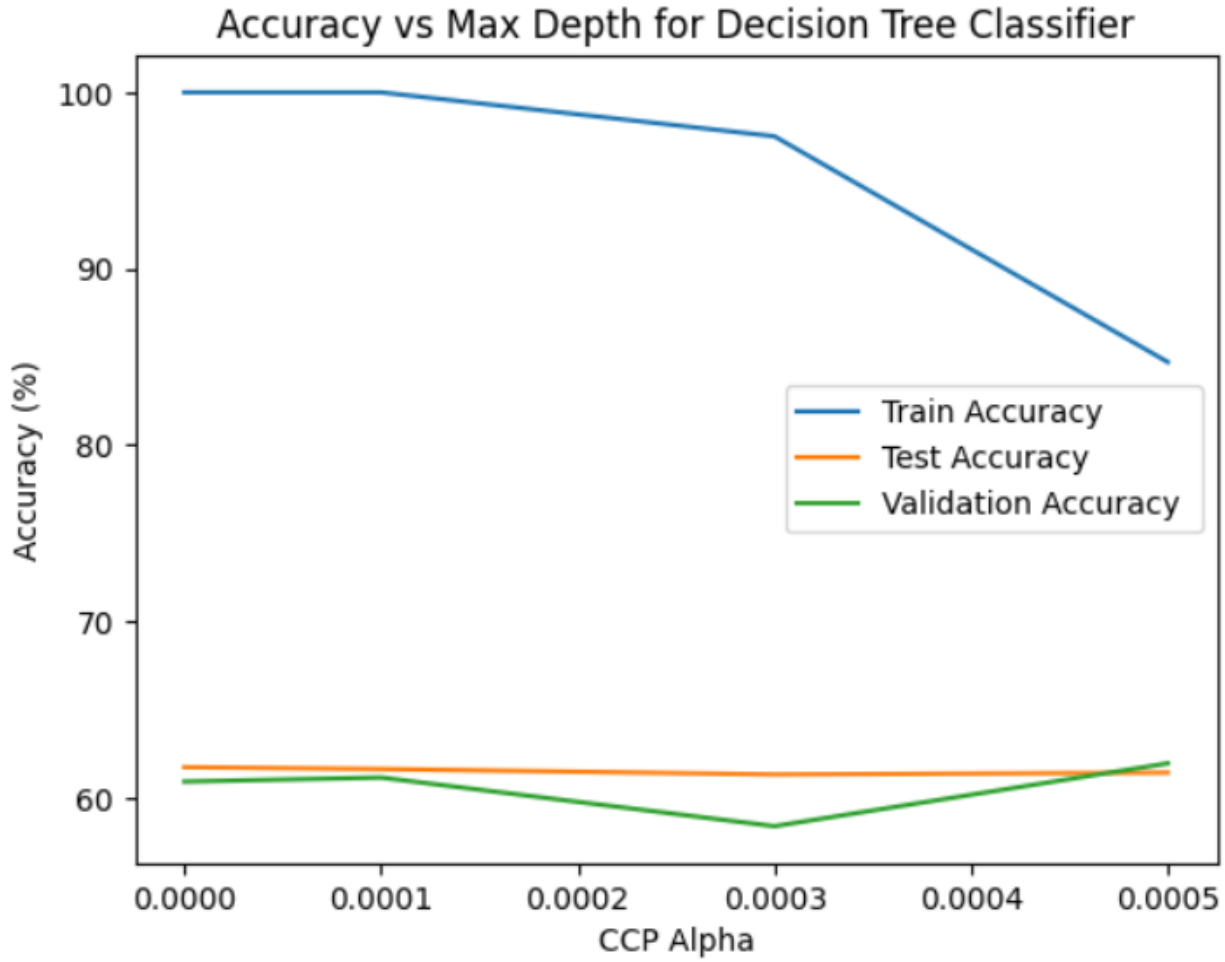
(E) (1)




```
Train-Accuracy, Validation-Accuracy and Test-Accuracy for depth 15 : 72.57%, 59.20%, 59.15%

Train-Accuracy, Validation-Accuracy and Test-Accuracy for depth 25 : 86.25%, 60.11%, 61.53%

Train-Accuracy, Validation-Accuracy and Test-Accuracy for depth 35 : 96.01%, 60.80%, 60.70%

Train-Accuracy, Validation-Accuracy and Test-Accuracy for depth 45 : 99.57%, 60.69%, 61.22%
```

Comments – Best Validation accuracy is for depth – 35 , therefore chosen depth is 35.

(E) (2)



Best validation accuracy is for ccp_alpha = 0.0005

Comments - These values are close to those in b and c, thus the my model works!

Pruning superiority over depth limiting: Post-pruning via CCP alpha demonstrates better generalization than pre-pruning through max_depth constraints. This is because CCP alpha allows the tree to initially grow fully and then removes subtrees that don't contribute significantly to model performance, resulting in a more optimal balance between bias and variance.

Gini vs Entropy: Across both pruning strategies, Gini impurity consistently yields slightly better validation accuracy than entropy (~1% improvement). This suggests that Gini's tendency toward creating purer nodes is beneficial for this particular dataset.

(F) Best Hyperparameters:

estimators: 150, max_features: 0.3, min_samples_split: 8 => Train Acc: 98.42851667305482 , Test Acc: 68.14891416752845 , Val Acc: 70.91954022988506

Out-of-bag accuracy (OOB Accuracy): 70.30790852178357 %

Comments – The accuracy of the random forest algorithm is significantly better than any of our simple tree algorithms for test and validation data, whereas the accuracy for our training data has decreased. This means that our previous models were overfitting which the bagging used in the random forest reduced.

DHRUV GUPTA

2023EE30858

COL 774

Assignment – 3 (Neural Network)

(B) Stopping Criteria = Max Epochs of 200.

Hidden Layer Size = 1

Accuracy:

Train Accuracy: 6.88%

Test Accuracy: 6.76%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.0639	0.1215	0.0433
2	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000
6	0.0746	0.0637	0.0900
7	0.0000	0.0000	0.0000
8	0.0404	0.0379	0.0433
9	0.0000	0.0000	0.0000
10	0.0079	0.0065	0.0100
11	0.0136	0.0211	0.0100
12	0.0125	0.0952	0.0067
13	0.0211	0.0286	0.0167
14	0.2050	0.1147	0.9600
15	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000
18	0.0494	0.0759	0.0367
19	0.0000	0.0000	0.0000
20	0.0352	0.0516	0.0267
21	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000
23	0.0354	0.0431	0.0300
24	0.1112	0.0592	0.9200

25	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000
27	0.1000	0.1438	0.0767
28	0.0412	0.0368	0.0467
29	0.0565	0.0436	0.0800
30	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000
33	0.0000	0.0000	0.0000
34	0.0000	0.0000	0.0000
35	0.0000	0.0000	0.0000
36	0.0351	0.0336	0.0367

Average (Macro) Metrics (Test):

F1: 0.0251, Precision: 0.0271, Recall: 0.0676

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.0323	0.0631	0.0217
2	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000
6	0.0761	0.0651	0.0917
7	0.0000	0.0000	0.0000
8	0.0511	0.0464	0.0567
9	0.0000	0.0000	0.0000
10	0.0191	0.0161	0.0233
11	0.0180	0.0277	0.0133
12	0.0032	0.0323	0.0017
13	0.0183	0.0236	0.0150
14	0.2091	0.1171	0.9717
15	0.0000	0.0000	0.0000

16	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000
18	0.0337	0.0515	0.0250
19	0.0000	0.0000	0.0000
20	0.0368	0.0525	0.0283
21	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000
23	0.0350	0.0458	0.0283
24	0.1121	0.0597	0.9200
25	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000
27	0.0477	0.0681	0.0367
28	0.0652	0.0558	0.0783
29	0.0757	0.0586	0.1067
30	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000
33	0.0000	0.0000	0.0000
34	0.0000	0.0000	0.0000
35	0.0095	0.0882	0.0050
36	0.0465	0.0423	0.0517

Average (Macro) Metrics (Train):

F1: 0.0247, Precision: 0.0254, Recall: 0.0688

Hidden Layer Size = 5

Accuracy:

Train Accuracy: 41.42%

Test Accuracy: 37.22%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.6033	0.5935	0.6133
2	0.0173	0.0652	0.0100
3	0.2609	0.3013	0.2300
4	0.2157	0.2619	0.1833
5	0.3541	0.4467	0.2933
6	0.3888	0.3483	0.4400
7	0.4103	0.3981	0.4233
8	0.5065	0.4135	0.6533
9	0.6500	0.5816	0.7367
10	0.5594	0.4371	0.7767
11	0.6144	0.5054	0.7833
12	0.5753	0.5772	0.5733
13	0.4618	0.4132	0.5233
14	0.4567	0.3975	0.5367
15	0.2015	0.2236	0.1833
16	0.3084	0.3006	0.3167
17	0.3140	0.2495	0.4233
18	0.2562	0.3004	0.2233
19	0.1589	0.2656	0.1133
20	0.3636	0.3099	0.4400
21	0.3899	0.3366	0.4633
22	0.5910	0.5245	0.6767
23	0.2119	0.3137	0.1600
24	0.3205	0.4464	0.2500
25	0.3041	0.3016	0.3067
26	0.0674	0.1512	0.0433

27	0.3343	0.2944	0.3867
28	0.2506	0.3810	0.1867
29	0.1474	0.2804	0.1000
30	0.4418	0.3634	0.5633
31	0.4011	0.3480	0.4733
32	0.2211	0.2190	0.2233
33	0.3213	0.3504	0.2967
34	0.3376	0.3232	0.3533
35	0.1091	0.2471	0.0700
36	0.3575	0.3458	0.3700

Average (Macro) Metrics (Test):

F1: 0.3468, Precision: 0.3505, Recall: 0.3722

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.6577	0.6605	0.6550
2	0.0184	0.1154	0.0100
3	0.2782	0.3060	0.2550
4	0.2318	0.2823	0.1967
5	0.3996	0.4915	0.3367
6	0.4417	0.3945	0.5017
7	0.5016	0.4816	0.5233
8	0.5690	0.4909	0.6767
9	0.6980	0.6172	0.8033
10	0.5552	0.4352	0.7667
11	0.6033	0.4867	0.7933
12	0.5987	0.5957	0.6017
13	0.5299	0.4755	0.5983
14	0.5071	0.4428	0.5933
15	0.2757	0.2894	0.2633
16	0.3508	0.3407	0.3617
17	0.3451	0.2780	0.4550

18	0.2644	0.2996	0.2367
19	0.1364	0.2534	0.0933
20	0.4225	0.3590	0.5133
21	0.4446	0.3803	0.5350
22	0.6822	0.6199	0.7583
23	0.2656	0.3891	0.2017
24	0.3785	0.4986	0.3050
25	0.3276	0.3236	0.3317
26	0.0780	0.1775	0.0500
27	0.3913	0.3567	0.4333
28	0.2568	0.4036	0.1883
29	0.2086	0.3953	0.1417
30	0.4246	0.3520	0.5350
31	0.4756	0.4219	0.5450
32	0.2790	0.2732	0.2850
33	0.3966	0.4547	0.3517
34	0.4048	0.3656	0.4533
35	0.1636	0.3706	0.1050
36	0.4111	0.3750	0.4550

Average (Macro) Metrics (Train):

F1: 0.3882, Precision: 0.3959, Recall: 0.4142

Hidden Layer Size = 10

Accuracy:

Train Accuracy: 61.58%

Test Accuracy: 54.92%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.6820	0.6299	0.7433
2	0.3519	0.3958	0.3167
3	0.6091	0.5583	0.6700
4	0.4924	0.5017	0.4833
5	0.4391	0.4917	0.3967
6	0.5887	0.6518	0.5367
7	0.4721	0.4645	0.4800
8	0.6667	0.6601	0.6733
9	0.7852	0.7905	0.7800
10	0.6646	0.6325	0.7000
11	0.7532	0.7169	0.7933
12	0.6556	0.6545	0.6567
13	0.5427	0.5559	0.5300
14	0.6364	0.6006	0.6767
15	0.5657	0.5226	0.6167
16	0.5836	0.5074	0.6867
17	0.4065	0.3968	0.4167
18	0.4441	0.4591	0.4300
19	0.5870	0.6429	0.5400
20	0.5442	0.5668	0.5233
21	0.5167	0.4954	0.5400
22	0.7488	0.7200	0.7800
23	0.5166	0.5785	0.4667
24	0.5997	0.5749	0.6267
25	0.4485	0.4901	0.4133
26	0.2922	0.3392	0.2567

27	0.6730	0.6424	0.7067
28	0.5284	0.5878	0.4800
29	0.4031	0.4815	0.3467
30	0.4724	0.4478	0.5000
31	0.5629	0.4925	0.6567
32	0.2944	0.3724	0.2433
33	0.4832	0.4463	0.5267
34	0.5637	0.5396	0.5900
35	0.4473	0.4920	0.4100
36	0.5518	0.5291	0.5767

Average (Macro) Metrics (Test):

F1: 0.5437, Precision: 0.5453, Recall: 0.5492

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.7486	0.7225	0.7767
2	0.4537	0.5104	0.4083
3	0.6840	0.6335	0.7433
4	0.5295	0.5273	0.5317
5	0.4688	0.5129	0.4317
6	0.6508	0.6930	0.6133
7	0.5483	0.5340	0.5633
8	0.7320	0.7357	0.7283
9	0.8122	0.8247	0.8000
10	0.7002	0.6939	0.7067
11	0.7885	0.7391	0.8450
12	0.7532	0.7615	0.7450
13	0.6289	0.6197	0.6383
14	0.7367	0.7254	0.7483
15	0.6533	0.6063	0.7083
16	0.6469	0.5716	0.7450
17	0.4996	0.4807	0.5200

18	0.4505	0.4902	0.4167
19	0.5973	0.6535	0.5500
20	0.6678	0.6723	0.6633
21	0.5828	0.5398	0.6333
22	0.7880	0.7686	0.8083
23	0.6013	0.6920	0.5317
24	0.6629	0.6365	0.6917
25	0.5507	0.5637	0.5383
26	0.3025	0.3584	0.2617
27	0.6997	0.6447	0.7650
28	0.6235	0.6764	0.5783
29	0.4814	0.5441	0.4317
30	0.5080	0.4671	0.5567
31	0.6347	0.5771	0.7050
32	0.3701	0.4704	0.3050
33	0.5731	0.5647	0.5817
34	0.6405	0.6053	0.6800
35	0.5657	0.6250	0.5167
36	0.6599	0.6228	0.7017

Average (Macro) Metrics (Train):

F1: 0.6110, Precision: 0.6129, Recall: 0.6158

Hidden layer size: 50

Accuracy:

Train Accuracy: 86.73%

Test Accuracy: 75.50%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.8221	0.8277	0.8167
2	0.6851	0.6677	0.7033
3	0.7871	0.8049	0.7700
4	0.6688	0.6550	0.6833
5	0.7255	0.7243	0.7267
6	0.7756	0.7680	0.7833
7	0.6966	0.7214	0.6733
8	0.8163	0.7968	0.8367
9	0.8803	0.8908	0.8700
10	0.8411	0.8355	0.8467
11	0.9151	0.9136	0.9167
12	0.8467	0.8467	0.8467
13	0.7496	0.7666	0.7333
14	0.8062	0.7536	0.8667
15	0.7902	0.7714	0.8100
16	0.8038	0.7706	0.8400
17	0.6644	0.6689	0.6600
18	0.7464	0.8038	0.6967
19	0.7038	0.6913	0.7167
20	0.7010	0.7234	0.6800
21	0.7273	0.7213	0.7333
22	0.8660	0.8494	0.8833
23	0.6839	0.7097	0.6600
24	0.7552	0.7821	0.7300
25	0.7155	0.7143	0.7167
26	0.5811	0.5890	0.5733

27 0.8286 0.8272 0.8300

28 0.8316 0.8582 0.8067

29 0.6387 0.6441 0.6333

30 0.7492 0.7517 0.7467

31 0.7649 0.7219 0.8133

32 0.6169 0.6276 0.6067

33 0.7182 0.7411 0.6967

34 0.7480 0.7302 0.7667

35 0.7533 0.7533 0.7533

36 0.7597 0.7661 0.7533

Average (Macro) Metrics (Test):

F1: 0.7545, Precision: 0.7553, Recall: 0.7550

Per-Class Metrics (Train):

Class F1 Precision Recall

1 0.9075 0.9240 0.8917

2 0.8200 0.8088 0.8317

3 0.8708 0.8767 0.8650

4 0.8184 0.8071 0.8300

5 0.8356 0.8412 0.8300

6 0.8742 0.8574 0.8917

7 0.8569 0.8656 0.8483

8 0.8941 0.8948 0.8933

9 0.9415 0.9446 0.9383

10 0.8904 0.9010 0.8800

11 0.9423 0.9316 0.9533

12 0.9172 0.9210 0.9133

13 0.8540 0.8650 0.8433

14 0.9053 0.8943 0.9167

15 0.8835 0.8585 0.9100

16 0.9064 0.9012 0.9117

17 0.7980 0.7896 0.8067

18	0.8705	0.8902	0.8517
19	0.8232	0.8316	0.8150
20	0.8731	0.8746	0.8717
21	0.8576	0.8472	0.8683
22	0.9323	0.9232	0.9417
23	0.8176	0.8492	0.7883
24	0.8733	0.8733	0.8733
25	0.8461	0.8540	0.8383
26	0.7721	0.7793	0.7650
27	0.9181	0.9211	0.9150
28	0.9092	0.9085	0.9100
29	0.8156	0.8129	0.8183
30	0.8622	0.8643	0.8600
31	0.8801	0.8565	0.9050
32	0.7633	0.7717	0.7550
33	0.8689	0.8942	0.8450
34	0.8946	0.8977	0.8917
35	0.8348	0.8233	0.8467
36	0.8889	0.8718	0.9067

Average (Macro) Metrics (Train):

F1: 0.8672, Precision: 0.8674, Recall: 0.8673

Hidden layer size: 100

Accuracy:

Train Accuracy: 92.93%

Test Accuracy: 81.36%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.8893	0.8820	0.8967
2	0.7174	0.7115	0.7233
3	0.8112	0.8529	0.7733
4	0.7480	0.7302	0.7667
5	0.7812	0.8152	0.7500
6	0.8255	0.8311	0.8200
7	0.7805	0.8175	0.7467
8	0.8339	0.8153	0.8533
9	0.8919	0.9041	0.8800
10	0.8767	0.9014	0.8533
11	0.9264	0.9295	0.9233
12	0.8930	0.9100	0.8767
13	0.7888	0.7810	0.7967
14	0.8640	0.8308	0.9000
15	0.8355	0.8247	0.8467
16	0.9034	0.8875	0.9200
17	0.7379	0.7391	0.7367
18	0.7864	0.8000	0.7733
19	0.7440	0.7196	0.7700
20	0.7614	0.7732	0.7500
21	0.8134	0.7798	0.8500
22	0.9016	0.8871	0.9167
23	0.7579	0.7592	0.7567
24	0.8062	0.8304	0.7833
25	0.7844	0.7993	0.7700
26	0.7267	0.7267	0.7267

27	0.8832	0.8969	0.8700
28	0.8816	0.9081	0.8567
29	0.7645	0.7832	0.7467
30	0.8084	0.7880	0.8300
31	0.8208	0.7768	0.8700
32	0.6940	0.7443	0.6500
33	0.7778	0.7424	0.8167
34	0.8271	0.8025	0.8533
35	0.8040	0.8013	0.8067
36	0.8356	0.8412	0.8300

Average (Macro) Metrics (Test):

F1: 0.8134, Precision: 0.8146, Recall: 0.8136

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.9502	0.9455	0.9550
2	0.9065	0.8998	0.9133
3	0.9266	0.9493	0.9050
4	0.8957	0.8965	0.8950
5	0.8923	0.9234	0.8633
6	0.9226	0.9121	0.9333
7	0.9347	0.9516	0.9183
8	0.9389	0.9297	0.9483
9	0.9716	0.9748	0.9683
10	0.9413	0.9476	0.9350
11	0.9624	0.9648	0.9600
12	0.9410	0.9386	0.9433
13	0.9019	0.8923	0.9117
14	0.9529	0.9458	0.9600
15	0.9395	0.9341	0.9450
16	0.9617	0.9617	0.9617
17	0.8953	0.8924	0.8983

18 0.9239 0.9381 0.9100

19 0.8979 0.8942 0.9017

20 0.9348 0.9501 0.9200

21 0.9237 0.8915 0.9583

22 0.9638 0.9528 0.9750

23 0.9067 0.9067 0.9067

24 0.9425 0.9432 0.9417

25 0.9147 0.9371 0.8933

26 0.8872 0.8828 0.8917

27 0.9593 0.9569 0.9617

28 0.9625 0.9617 0.9633

29 0.9086 0.9056 0.9117

30 0.9336 0.9192 0.9483

31 0.9299 0.9201 0.9400

32 0.8960 0.9091 0.8833

33 0.9350 0.9350 0.9350

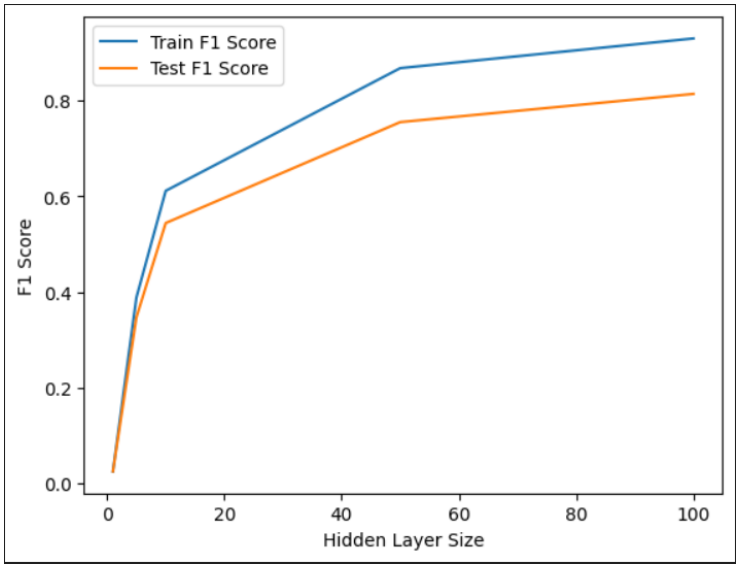
34 0.9386 0.9340 0.9433

35 0.9146 0.9280 0.9017

36 0.9430 0.9345 0.9517

Average (Macro) Metrics (Train):

F1: 0.9292, Precision: 0.9295, Recall: 0.9293



Comments – Stopping criteria is 200 epochs ,The experimental results show that increasing the number of hidden units from 1 to 100 led to a consistent improvement in F1 score, precision, recall, and accuracy for both training and test datasets. While larger hidden layers enhanced the model's representational capacity and overall performance, the widening gap between training and test accuracies at higher unit counts indicates the onset of mild overfiting.

(C)

=== Hidden layer architecture: [512] ===

Accuracy:

Train Accuracy: 72.20%

Test Accuracy: 69.52%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.7809	0.7720	0.7900
2	0.6244	0.6431	0.6067
3	0.7119	0.7241	0.7000
4	0.5850	0.5737	0.5967
5	0.6298	0.6547	0.6067
6	0.7427	0.7261	0.7600
7	0.6387	0.6703	0.6100
8	0.7869	0.7742	0.8000
9	0.8446	0.8562	0.8333
10	0.8345	0.8459	0.8233
11	0.8754	0.8613	0.8900
12	0.7961	0.7857	0.8067
13	0.6818	0.7169	0.6500
14	0.7929	0.7704	0.8167
15	0.7164	0.6972	0.7367
16	0.7346	0.6839	0.7933
17	0.6072	0.6254	0.5900
18	0.6775	0.7373	0.6267
19	0.6769	0.6910	0.6633
20	0.6586	0.6821	0.6367
21	0.6093	0.5162	0.7433
22	0.8114	0.8197	0.8033
23	0.6629	0.7675	0.5833
24	0.7138	0.7594	0.6733

25	0.6149	0.6098	0.6200
26	0.4708	0.5411	0.4167
27	0.7516	0.7281	0.7767
28	0.7980	0.8131	0.7833
29	0.5244	0.5731	0.4833
30	0.6370	0.5733	0.7167
31	0.6965	0.6476	0.7533
32	0.5377	0.5528	0.5233
33	0.6634	0.6506	0.6767
34	0.7367	0.6953	0.7833
35	0.6273	0.6348	0.6200
36	0.7321	0.7309	0.7333

Average (Macro) Metrics (Test):

F1: 0.6940, Precision: 0.6974, Recall: 0.6952

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.8033	0.7967	0.8100
2	0.6386	0.6601	0.6183
3	0.7523	0.7633	0.7417
4	0.6384	0.6489	0.6283
5	0.6517	0.7084	0.6033
6	0.7631	0.7484	0.7783
7	0.7153	0.7367	0.6950
8	0.8212	0.8311	0.8117
9	0.8684	0.8912	0.8467
10	0.8063	0.8043	0.8083
11	0.8430	0.8192	0.8683
12	0.8133	0.8133	0.8133
13	0.7094	0.7088	0.7100
14	0.8299	0.8185	0.8417
15	0.7214	0.6978	0.7467

16 0.7406 0.7147 0.7683

17 0.6328 0.6536 0.6133

18 0.6937 0.7373 0.6550

19 0.7162 0.7260 0.7067

20 0.7379 0.7391 0.7367

21 0.6338 0.5330 0.7817

22 0.8330 0.8519 0.8150

23 0.6856 0.7908 0.6050

24 0.7320 0.7553 0.7100

25 0.6405 0.6282 0.6533

26 0.4930 0.5605 0.4400

27 0.7964 0.7727 0.8217

28 0.8302 0.8254 0.8350

29 0.5746 0.5919 0.5583

30 0.6755 0.6047 0.7650

31 0.7119 0.6830 0.7433

32 0.5819 0.6240 0.5450

33 0.6699 0.6540 0.6867

34 0.7700 0.7540 0.7867

35 0.6644 0.6776 0.6517

36 0.7705 0.7504 0.7917

Average (Macro) Metrics (Train):

F1: 0.7211, Precision: 0.7243, Recall: 0.7220

==========

=== Hidden layer architecture: [512, 256] ===

Accuracy:

Train Accuracy: 63.83%

Test Accuracy: 62.64%

Per-Class Metrics (Test):

Class F1 Precision Recall

1	0.7198	0.6657	0.7833
2	0.5226	0.6828	0.4233
3	0.6774	0.6533	0.7033
4	0.5291	0.4887	0.5767
5	0.5675	0.5825	0.5533
6	0.7030	0.6780	0.7300
7	0.5473	0.5658	0.5300
8	0.7298	0.6831	0.7833
9	0.7776	0.7289	0.8333
10	0.7783	0.7462	0.8133
11	0.7922	0.7722	0.8133
12	0.7422	0.7314	0.7533
13	0.5521	0.6560	0.4767
14	0.6927	0.6294	0.7700
15	0.6431	0.6497	0.6367
16	0.6686	0.5888	0.7733
17	0.5073	0.4952	0.5200
18	0.5476	0.6558	0.4700
19	0.6284	0.6370	0.6200
20	0.6200	0.6531	0.5900
21	0.5705	0.5677	0.5733
22	0.7531	0.7088	0.8033
23	0.5974	0.6586	0.5467
24	0.6620	0.6934	0.6333
25	0.4806	0.5741	0.4133
26	0.3553	0.4428	0.2967
27	0.7044	0.6834	0.7267
28	0.7161	0.7826	0.6600
29	0.4247	0.4366	0.4133
30	0.6362	0.6230	0.6500
31	0.6516	0.5665	0.7667
32	0.4348	0.5022	0.3833
33	0.5487	0.4986	0.6100
34	0.6425	0.5868	0.7100

35	0.5650	0.5884	0.5433
36	0.6711	0.6757	0.6667

Average (Macro) Metrics (Test):

F1: 0.6211, Precision: 0.6259, Recall: 0.6264

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.7529	0.7072	0.8050
2	0.5058	0.6928	0.3983
3	0.6631	0.6169	0.7167
4	0.5113	0.4789	0.5483
5	0.5400	0.5577	0.5233
6	0.7067	0.6722	0.7450
7	0.6050	0.6226	0.5883
8	0.7621	0.7406	0.7850
9	0.7895	0.7542	0.8283
10	0.7443	0.7010	0.7933
11	0.7814	0.7519	0.8133
12	0.7526	0.7330	0.7733
13	0.6072	0.6776	0.5500
14	0.7153	0.6756	0.7600
15	0.6420	0.6341	0.6500
16	0.6315	0.5763	0.6983
17	0.4918	0.4839	0.5000
18	0.5453	0.6461	0.4717
19	0.6293	0.6408	0.6183
20	0.6810	0.6771	0.6850
21	0.5740	0.5603	0.5883
22	0.7697	0.7239	0.8217
23	0.6249	0.6750	0.5817
24	0.6509	0.6870	0.6183
25	0.5326	0.6144	0.4700

26	0.3067	0.4055	0.2467
27	0.7427	0.7034	0.7867
28	0.7546	0.7866	0.7250
29	0.4812	0.4727	0.4900
30	0.6129	0.5997	0.6267
31	0.6553	0.5891	0.7383
32	0.4701	0.5571	0.4067
33	0.5732	0.5406	0.6100
34	0.6801	0.6340	0.7333
35	0.5803	0.6205	0.5450
36	0.7168	0.6965	0.7383

Average (Macro) Metrics (Train):

F1: 0.6329, Precision: 0.6363, Recall: 0.6383

==========

=== Hidden layer architecture: [512, 256, 128] ===

Accuracy:

Train Accuracy: 29.75%

Test Accuracy: 28.66%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.4420	0.6689	0.3300
2	0.1133	0.1635	0.0867
3	0.1316	0.2737	0.0867
4	0.2163	0.2311	0.2033
5	0.0475	0.2162	0.0267
6	0.4178	0.4977	0.3600
7	0.2864	0.1898	0.5833
8	0.4476	0.7833	0.3133
9	0.4182	0.3426	0.5367
10	0.4725	0.3281	0.8433

11	0.3995	0.3297	0.5067
12	0.2998	0.2964	0.3033
13	0.2098	0.3488	0.1500
14	0.3783	0.2630	0.6733
15	0.2130	0.3485	0.1533
16	0.2094	0.2727	0.1700
17	0.0065	0.1250	0.0033
18	0.2462	0.2000	0.3200
19	0.3120	0.2016	0.6900
20	0.3356	0.3451	0.3267
21	0.1857	0.1625	0.2167
22	0.3911	0.4028	0.3800
23	0.1798	0.1521	0.2200
24	0.4391	0.4523	0.4267
25	0.0633	0.1519	0.0400
26	0.0000	0.0000	0.0000
27	0.0725	0.3871	0.0400
28	0.4028	0.3085	0.5800
29	0.0544	0.2903	0.0300
30	0.2522	0.2657	0.2400
31	0.3363	0.3060	0.3733
32	0.0066	1.0000	0.0033
33	0.2474	0.3333	0.1967
34	0.2474	0.2259	0.2733
35	0.2952	0.4351	0.2233
36	0.3620	0.3262	0.4067

Average (Macro) Metrics (Test):

F1: 0.2539, Precision: 0.3229, Recall: 0.2866

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.4479	0.6826	0.3333

2	0.1581	0.2202	0.1233
3	0.1416	0.2932	0.0933
4	0.2170	0.2264	0.2083
5	0.0503	0.2237	0.0283
6	0.4349	0.5273	0.3700
7	0.2689	0.1781	0.5483
8	0.4906	0.8387	0.3467
9	0.4601	0.3712	0.6050
10	0.4667	0.3267	0.8167
11	0.4113	0.3354	0.5317
12	0.3189	0.3130	0.3250
13	0.2150	0.3509	0.1550
14	0.4067	0.2816	0.7317
15	0.2404	0.4008	0.1717
16	0.2152	0.2931	0.1700
17	0.0195	0.4286	0.0100
18	0.2519	0.2031	0.3317
19	0.3203	0.2078	0.6983
20	0.3782	0.3798	0.3767
21	0.2199	0.1914	0.2583
22	0.4275	0.4352	0.4200
23	0.1990	0.1717	0.2367
24	0.4215	0.4394	0.4050
25	0.0883	0.2000	0.0567
26	0.0000	0.0000	0.0000
27	0.0470	0.3947	0.0250
28	0.4408	0.3314	0.6583
29	0.0611	0.3636	0.0333
30	0.2051	0.2251	0.1883
31	0.3272	0.3023	0.3567
32	0.0000	0.0000	0.0000
33	0.2939	0.3834	0.2383
34	0.2539	0.2396	0.2700
35	0.2457	0.4030	0.1767

36 0.3517 0.3079 0.4100

Average (Macro) Metrics (Train):

F1: 0.2638, Precision: 0.3186, Recall: 0.2975

==========

=== Hidden layer architecture: [512, 256, 128, 64] ===

Accuracy:

Train Accuracy: 3.25%

Test Accuracy: 3.27%

Per-Class Metrics (Test):

Class	F1	Precision	Recall
1	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000
12	0.0620	0.0320	1.0000
13	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000

21	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000
23	0.0000	0.0000	0.0000
24	0.0066	0.5000	0.0033
25	0.0605	0.0366	0.1733
26	0.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000
28	0.0000	0.0000	0.0000
29	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000
33	0.0000	0.0000	0.0000
34	0.0000	0.0000	0.0000
35	0.0000	0.0000	0.0000
36	0.0000	0.0000	0.0000

Average (Macro) Metrics (Test):

F1: 0.0036, Precision: 0.0158, Recall: 0.0327

Per-Class Metrics (Train):

Class	F1	Precision	Recall
1	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000

12 0.0617 0.0318 1.0000

13 0.0000 0.0000 0.0000

14 0.0000 0.0000 0.0000

15 0.0000 0.0000 0.0000

16 0.0000 0.0000 0.0000

17 0.0000 0.0000 0.0000

18 0.0000 0.0000 0.0000

19 0.0000 0.0000 0.0000

20 0.0000 0.0000 0.0000

21 0.0000 0.0000 0.0000

22 0.0000 0.0000 0.0000

23 0.0000 0.0000 0.0000

24 0.0000 0.0000 0.0000

25 0.0605 0.0369 0.1683

26 0.0000 0.0000 0.0000

27 0.0000 0.0000 0.0000

28 0.0000 0.0000 0.0000

29 0.0000 0.0000 0.0000

30 0.0000 0.0000 0.0000

31 0.0000 0.0000 0.0000

32 0.0000 0.0000 0.0000

33 0.0000 0.0000 0.0000

34 0.0000 0.0000 0.0000

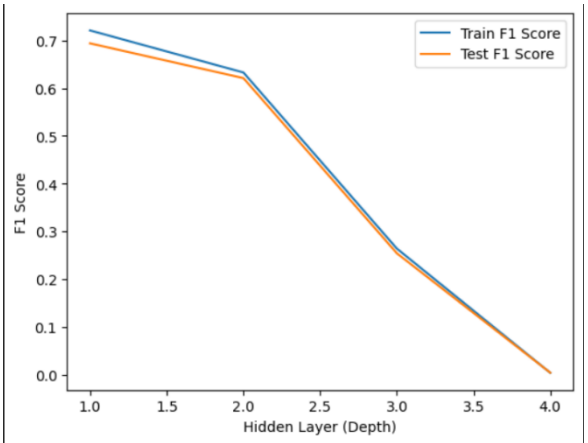
35 0.0000 0.0000 0.0000

36 0.0000 0.0000 0.0000

Average (Macro) Metrics (Train):

F1: 0.0034, Precision: 0.0019, Recall: 0.0325

==========



Here 1 = [512] , 2 =[512,256] ……. 4 = [512,256,128,64]

Comments – The deeper networks do not perform as the suffer from vanishing gradients for the deeper layers. Thus, we see the best performance across all the architectures when the hidden layers = {512}. The 4-layer network completely fails to predict accurately. Max epochs were 20 for this case , on increasing the number of epochs we see increase in accuracies over test data.

(D)

========= Training Model with Hidden Layers: [512] =========

Train Accuracy: 94.24%

Test Accuracy: 85.19%

Test Macro Metrics:

F1 Score: 0.8517

Precision: 0.8548

Recall: 0.8519

Train Macro Metrics:

F1 Score: 0.9424

Precision: 0.9434

Recall: 0.9424

Detailed Per-Class Metrics (Test Data):

	precision	recall	f1-score
0	0.9516	0.9167	0.9338
1	0.7720	0.8467	0.8076
2	0.8439	0.8467	0.8453
3	0.7470	0.8267	0.7848
4	0.8423	0.8367	0.8395
5	0.8235	0.8867	0.8539
6	0.8810	0.7900	0.8330
7	0.8955	0.8567	0.8756
8	0.9386	0.9167	0.9275
9	0.9397	0.8833	0.9107
10	0.9754	0.9267	0.9504
11	0.9153	0.9000	0.9076
12	0.7764	0.8567	0.8146
13	0.8313	0.9200	0.8734
14	0.8581	0.8867	0.8721

15	0.8675	0.9167	0.8914
16	0.8014	0.7800	0.7905
17	0.8938	0.8133	0.8517
18	0.8410	0.7933	0.8165
19	0.8433	0.7533	0.7958
20	0.8111	0.8733	0.8411
21	0.9228	0.9167	0.9197
22	0.8421	0.7467	0.7915
23	0.8493	0.8267	0.8378
24	0.7543	0.8700	0.8080
25	0.8315	0.7567	0.7923
26	0.9231	0.9200	0.9215
27	0.9199	0.8800	0.8995
28	0.7493	0.8367	0.7906
29	0.8716	0.8600	0.8658
30	0.7705	0.9067	0.8331
31	0.8761	0.6367	0.7375
32	0.8576	0.8633	0.8605
33	0.8517	0.9000	0.8752
34	0.8459	0.8600	0.8529
35	0.8571	0.8600	0.8586
accuracy		0.8519	
macro avg	0.8548	0.8519	0.8517
weighted avg	0.8548	0.8519	0.8517

Detailed Per-Class Metrics (Train Data):

	precision	recall	f1-score
0	0.9898	0.9667	0.9781
1	0.8988	0.9617	0.9291
2	0.9566	0.9550	0.9558
3	0.9013	0.9283	0.9146

4	0.9369	0.9150	0.9258
5	0.9233	0.9633	0.9429
6	0.9699	0.9117	0.9399
7	0.9571	0.9300	0.9434
8	0.9883	0.9817	0.9849
9	0.9544	0.9417	0.9480
10	0.9813	0.9600	0.9705
11	0.9681	0.9600	0.9640
12	0.9060	0.9317	0.9187
13	0.9184	0.9750	0.9458
14	0.9570	0.9650	0.9610
15	0.9570	0.9633	0.9601
16	0.9448	0.9133	0.9288
17	0.9545	0.9083	0.9308
18	0.9410	0.9300	0.9355
19	0.9634	0.9217	0.9421
20	0.9088	0.9633	0.9353
21	0.9816	0.9767	0.9791
22	0.9468	0.8900	0.9175
23	0.9410	0.9567	0.9488
24	0.8873	0.9583	0.9215
25	0.9488	0.8950	0.9211
26	0.9639	0.9800	0.9719
27	0.9775	0.9417	0.9593
28	0.8799	0.9283	0.9035
29	0.9579	0.9483	0.9531
30	0.8848	0.9733	0.9270
31	0.9725	0.8267	0.8937
32	0.9277	0.9617	0.9444
33	0.9642	0.9417	0.9528
34	0.9179	0.9317	0.9247
35	0.9357	0.9700	0.9525

accuracy	0.9424

macro avg 0.9434 0.9424 0.9424

weighted avg 0.9434 0.9424 0.9424

==========

========= Model with Hidden Layers: [512, 256] =========

Train Accuracy: 98.25%

Test Accuracy: 87.47%

Test Macro Metrics:

F1 Score: 0.8746

Precision: 0.8767

Recall: 0.8747

Train Macro Metrics:

F1 Score: 0.9825

Precision: 0.9827

Recall: 0.9825

Detailed Per-Class Metrics (Test Data):

	precision	recall	f1-score	support
0	0.9398	0.9367	0.9382	
1	0.8328	0.8633	0.8478	
2	0.9161	0.8733	0.8942	
3	0.7932	0.8567	0.8237	
4	0.8547	0.8433	0.8490	
5	0.9075	0.8833	0.8953	
6	0.8431	0.8600	0.8515	
7	0.8442	0.9033	0.8728	
8	0.9191	0.9467	0.9327	
9	0.9100	0.9433	0.9264	
10	0.9720	0.9267	0.9488	
11	0.8743	0.9733	0.9211	

12	0.8416	0.8500	0.8458
13	0.8673	0.9367	0.9006
14	0.9331	0.8833	0.9075
15	0.9094	0.9033	0.9064
16	0.7880	0.8300	0.8084
17	0.8851	0.8733	0.8792
18	0.8893	0.7233	0.7978
19	0.7903	0.8667	0.8267
20	0.8119	0.9067	0.8567
21	0.9094	0.9367	0.9228
22	0.8381	0.7767	0.8062
23	0.8836	0.8600	0.8716
24	0.8600	0.8600	0.8600
25	0.8505	0.7967	0.8227
26	0.9137	0.9533	0.9331
27	0.9455	0.8667	0.9043
28	0.7435	0.8600	0.7975
29	0.8829	0.9300	0.9058
30	0.9158	0.8700	0.8923
31	0.8364	0.7667	0.8000
32	0.9275	0.8100	0.8648
33	0.9076	0.9167	0.9121
34	0.9206	0.8500	0.8839
35	0.9046	0.8533	0.8782
accuracy			0.8747
macro avg	0.8767	0.8747	0.8746
weighted avg	0.8767	0.8747	0.8746

Detailed Per-Class Metrics (Train Data):

	precision	recall	f1-score
0	0.9900	0.9900	0.9900
1	0.9818	0.9917	0.9867

2	0.9917	0.9900	0.9908
3	0.9686	0.9783	0.9735
4	0.9751	0.9783	0.9767
5	0.9949	0.9850	0.9899
6	0.9689	0.9867	0.9777
7	0.9674	0.9900	0.9786
8	0.9934	0.9967	0.9950
9	0.9787	0.9950	0.9868
10	1.0000	0.9800	0.9899
11	0.9706	0.9917	0.9810
12	0.9799	0.9750	0.9774
13	0.9642	0.9867	0.9753
14	0.9966	0.9900	0.9933
15	0.9866	0.9850	0.9858
16	0.9611	0.9883	0.9745
17	0.9882	0.9733	0.9807
18	0.9982	0.9483	0.9726
19	0.9735	0.9783	0.9759
20	0.9692	0.9967	0.9827
21	0.9852	0.9983	0.9917
22	0.9812	0.9567	0.9688
23	0.9916	0.9817	0.9866
24	0.9785	0.9883	0.9834
25	0.9814	0.9683	0.9748
26	0.9820	1.0000	0.9909
27	0.9983	0.9617	0.9796
28	0.9281	0.9900	0.9581
29	0.9868	0.9967	0.9917
30	1.0000	0.9733	0.9865
31	0.9898	0.9733	0.9815
32	0.9932	0.9783	0.9857
33	0.9933	0.9900	0.9917
34	0.9966	0.9650	0.9805
35	0.9932	0.9717	0.9823

accuracy			0.9825
macro avg	0.9827	0.9825	0.9825
weighted avg	0.9827	0.9825	0.9825

==========

========= Model with Hidden Layers: [512, 256, 128] =========

Train Accuracy: 99.08%

Test Accuracy: 87.66%

Test Macro Metrics:

F1 Score: 0.8768

Precision: 0.8813

Recall: 0.8766

Train Macro Metrics:

F1 Score: 0.9909

Precision: 0.9917

Recall: 0.9908

Detailed Per-Class Metrics (Test Data):

	precision	recall	f1-score
0	0.9191	0.9467	0.9327
1	0.8543	0.8600	0.8571
2	0.8307	0.8833	0.8562
3	0.9015	0.7933	0.8440
4	0.8815	0.7933	0.8351
5	0.8787	0.8933	0.8860
6	0.9148	0.8233	0.8667
7	0.8833	0.8833	0.8833
8	0.9530	0.9467	0.9498
9	0.9329	0.9267	0.9298

10	0.9661	0.9500	0.9580
11	0.9293	0.9200	0.9246
12	0.8397	0.8733	0.8562
13	0.9072	0.8800	0.8934
14	0.9249	0.9033	0.9140
15	0.9246	0.9400	0.9322
16	0.6303	0.8867	0.7368
17	0.8198	0.9100	0.8626
18	0.8704	0.7167	0.7861
19	0.8174	0.9100	0.8612
20	0.8447	0.9067	0.8746
21	0.9621	0.9300	0.9458
22	0.8571	0.7600	0.8057
23	0.8721	0.8633	0.8677
24	0.8185	0.8867	0.8512
25	0.8333	0.8333	0.8333
26	0.9406	0.9500	0.9453
27	0.9228	0.9167	0.9197
28	0.7701	0.8933	0.8272
29	0.8854	0.8500	0.8673
30	0.8874	0.8933	0.8904
31	0.9352	0.6733	0.7829
32	0.9062	0.8700	0.8878
33	0.8977	0.9067	0.9022
34	0.9125	0.9033	0.9079
35	0.9010	0.8800	0.8904
accuracy		0.8766	
macro avg	0.8813	0.8766	0.8768
weighted avg	0.8813	0.8766	0.8768

Detailed Per-Class Metrics (Train Data):

	precision	recall	f1-score	support

0	1.0000	0.9983	0.9992
1	0.9934	0.9983	0.9958
2	0.9917	0.9967	0.9942
3	1.0000	0.9617	0.9805
4	1.0000	0.9933	0.9967
5	0.9967	0.9983	0.9975
6	1.0000	0.9950	0.9975
7	0.9983	0.9933	0.9958
8	1.0000	1.0000	1.0000
9	0.9950	0.9983	0.9967
10	1.0000	0.9983	0.9992
11	1.0000	0.9950	0.9975
12	0.9934	0.9983	0.9958
13	0.9983	0.9767	0.9874
14	0.9967	0.9933	0.9950
15	0.9967	1.0000	0.9983
16	0.8403	1.0000	0.9132
17	0.9756	0.9983	0.9868
18	0.9962	0.8700	0.9288
19	0.9884	0.9983	0.9934
20	0.9917	0.9967	0.9942
21	0.9983	0.9983	0.9983
22	0.9966	0.9917	0.9942
23	0.9983	0.9950	0.9967
24	0.9917	0.9983	0.9950
25	0.9916	0.9883	0.9900
26	0.9983	1.0000	0.9992
27	0.9983	0.9883	0.9933
28	0.9788	0.9983	0.9884
29	1.0000	0.9983	0.9992
30	0.9983	0.9883	0.9933
31	1.0000	0.9733	0.9865
32	1.0000	1.0000	1.0000

33	1.0000	0.9967	0.9983
34	1.0000	0.9950	0.9975
35	0.9967	1.0000	0.9983
accuracy		0.9908	
macro avg	0.9917	0.9908	0.9909
weighted avg	0.9917	0.9908	0.9909

==========

Training Model with Hidden Layers: [512, 256, 128, 64]

Train Accuracy: 99.92%

Test Accuracy: 88.95%

Test Macro Metrics:

F1 Score: 0.8895

Precision: 0.8910

Recall: 0.8895

Train Macro Metrics:

F1 Score: 0.9992

Precision: 0.9992

Recall: 0.9992

Detailed Per-Class Metrics (Test Data):

	precision	recall	f1-score
0	0.9156	0.9400	0.9276
1	0.8138	0.9033	0.8562
2	0.8750	0.9100	0.8922
3	0.7859	0.8567	0.8198
4	0.8733	0.8500	0.8615
5	0.9033	0.9033	0.9033
6	0.9149	0.8600	0.8866

7	0.8822	0.9233	0.9023
8	0.9685	0.9233	0.9454
9	0.9248	0.9433	0.9340
10	0.9600	0.9600	0.9600
11	0.9426	0.9300	0.9362
12	0.8682	0.9000	0.8838
13	0.9307	0.9400	0.9353
14	0.8958	0.9167	0.9061
15	0.9307	0.9400	0.9353
16	0.8782	0.7933	0.8336
17	0.8767	0.8767	0.8767
18	0.8147	0.8500	0.8320
19	0.8553	0.8667	0.8609
20	0.8795	0.9000	0.8896
21	0.9532	0.9500	0.9516
22	0.9087	0.7300	0.8096
23	0.9046	0.8533	0.8782
24	0.8935	0.7833	0.8348
25	0.8630	0.8400	0.8514
26	0.9461	0.9367	0.9414
27	0.9490	0.9300	0.9394
28	0.8280	0.8667	0.8469
29	0.9016	0.9167	0.9091
30	0.8626	0.9000	0.8809
31	0.7781	0.8767	0.8245
32	0.8750	0.8867	0.8808
33	0.8937	0.8967	0.8952
34	0.9144	0.8900	0.9020
35	0.9135	0.8800	0.8964
accuracy		0.8895	
macro avg	0.8910	0.8895	0.8895
weighted avg	0.8910	0.8895	0.8895

Detailed Per-Class Metrics (Train Data):

	precision	recall	f1-score
0	1.0000	1.0000	1.0000
1	0.9983	1.0000	0.9992
2	0.9983	1.0000	0.9992
3	1.0000	0.9983	0.9992
4	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000
6	1.0000	1.0000	1.0000
7	0.9950	1.0000	0.9975
8	1.0000	1.0000	1.0000
9	1.0000	1.0000	1.0000
10	1.0000	1.0000	1.0000
11	1.0000	1.0000	1.0000
12	1.0000	1.0000	1.0000
13	0.9983	0.9983	0.9983
14	0.9983	0.9983	0.9983
15	1.0000	1.0000	1.0000
16	1.0000	1.0000	1.0000
17	0.9983	0.9983	0.9983
18	1.0000	1.0000	1.0000
19	0.9983	0.9983	0.9983
20	0.9983	1.0000	0.9992
21	1.0000	1.0000	1.0000
22	1.0000	0.9967	0.9983
23	1.0000	1.0000	1.0000
24	1.0000	0.9900	0.9950
25	1.0000	1.0000	1.0000
26	1.0000	1.0000	1.0000
27	1.0000	0.9933	0.9967
28	0.9967	1.0000	0.9983
29	0.9983	1.0000	0.9992

30	1.0000	0.9983	0.9992
31	0.9917	1.0000	0.9959
32	1.0000	1.0000	1.0000
33	1.0000	1.0000	1.0000
34	1.0000	1.0000	1.0000
35	1.0000	1.0000	1.0000

accuracy		0.9992	
macro avg	0.9992	0.9992	0.9992
weighted avg	0.9992	0.9992	0.9992

========== AVERAGE PERFORMANCE ACROSS ALL MODELS ==========

Average Train Accuracy: 97.87%

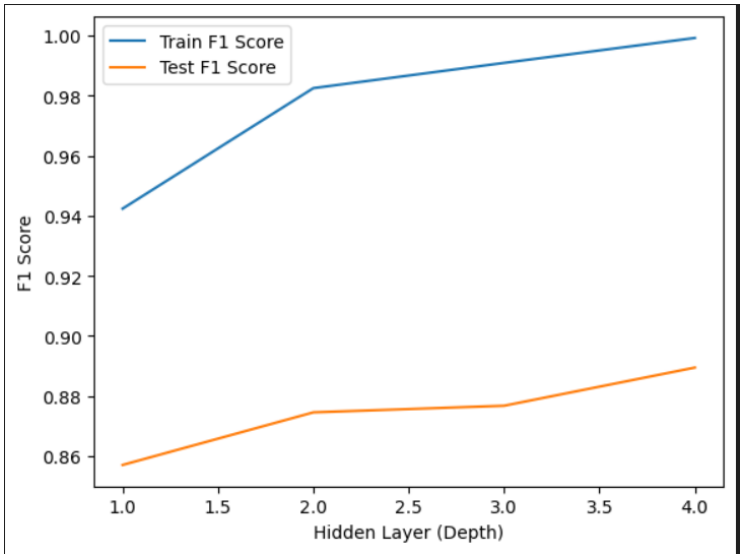
Average Test Accuracy: 87.32%

Average Test F1 Score: 0.8731

Average Test Precision: 0.8759

Average Test Recall: 0.8732

==========



Here 1 = [512] , 2 =[512,256] ……. 4 = [512,256,128,64]

Comments and comparison of test accuracies with part c– We see an overall increase in the efficiency of the model. This can be credited to the vanishing gradient problem. The relu solves the vanishing gradient problem for the 3-layer network as we can see from the improvement in the model's metric score. Thus the 3rd model performs the best. The accuracy of the 4th model has also increased considerably.

Depth	Sigmoid	Relu
512	69.5	86.19
512,256	62.5	87.4
512,256,128	28.8	87.6
512,256,128,64	3.8	88.9

Significant increase in accuracies for relu for same total number of epochs

(E)

Stopping Criteria -> Maximum epochs of 50.

========= Training MLP with Hidden Layers: [512] =========

Train Accuracy: 100.00%

Test Accuracy: 90.88%

Test F1 Score per Class:

Class 1: 0.9585

Class 2: 0.9014

Class 3: 0.9226

Class 4: 0.8514

Class 5: 0.8854

Class 6: 0.9160

Class 7: 0.9000

Class 8: 0.9130

Class 9: 0.9548

Class 10: 0.9451

Class 11: 0.9699

Class 12: 0.9456

Class 13: 0.9106

Class 14: 0.9243

Class 15: 0.9106

Class 16: 0.9360

Class 17: 0.8378

Class 18: 0.8988

Class 19: 0.8633

Class 20: 0.8827

Class 21: 0.8773

Class 22: 0.9567

Class 23: 0.8590

Class 24: 0.9079

Class 25: 0.8819

Class 26: 0.8591

Class 27: 0.9577

Class 28: 0.9394

Class 29: 0.8800

Class 30: 0.9076

Class 31: 0.9124

Class 32: 0.8635

Class 33: 0.9133

Class 34: 0.9294

Class 35: 0.9345

Class 36: 0.9067

Test Mean F1 Score: 0.9087

Test Precision (macro): 0.9090

Test Recall (macro): 0.9088

Train F1 Score per Class:

Class 1: 1.0000

Class 2: 1.0000

Class 3: 1.0000

Class 4: 1.0000

Class 5: 1.0000

Class 6: 1.0000

Class 7: 1.0000

Class 8: 1.0000

Class 9: 1.0000

Class 10: 1.0000

Class 11: 1.0000

Class 12: 1.0000

Class 13: 1.0000

Class 14: 1.0000

Class 15: 1.0000

Class 16: 1.0000

Class 17: 1.0000

Class 18: 1.0000

Class 19: 1.0000

Class 20: 1.0000

Class 21: 1.0000

Class 22: 1.0000

Class 23: 1.0000

Class 24: 1.0000

Class 25: 1.0000

Class 26: 1.0000

Class 27: 1.0000

Class 28: 1.0000

Class 29: 1.0000

Class 30: 1.0000

Class 31: 1.0000

Class 32: 1.0000

Class 33: 1.0000

Class 34: 1.0000

Class 35: 1.0000

Class 36: 1.0000

Train Mean F1 Score: 1.0000

Train Precision (macro): 1.0000

Train Recall (macro): 1.0000

========= Training MLP with Hidden Layers: [512, 256] =========

Train Accuracy: 100.00%

Test Accuracy: 91.65%

Test F1 Score per Class:

Class 1: 0.9715

Class 2: 0.8923

Class 3: 0.9428

Class 4: 0.8788

Class 5: 0.8896

Class 6: 0.9285

Class 7: 0.9112

Class 8: 0.9264

Class 9: 0.9583

Class 10: 0.9552

Class 11: 0.9651

Class 12: 0.9502

Class 13: 0.9073

Class 14: 0.9351

Class 15: 0.9311

Class 16: 0.9574

Class 17: 0.8543

Class 18: 0.8878

Class 19: 0.8686

Class 20: 0.9127

Class 21: 0.9034

Class 22: 0.9468

Class 23: 0.8656

Class 24: 0.9008

Class 25: 0.8959

Class 26: 0.8690

Class 27: 0.9712

Class 28: 0.9516

Class 29: 0.8822

Class 30: 0.9269

Class 31: 0.8918

Class 32: 0.8699

Class 33: 0.9057

Class 34: 0.9226

Class 35: 0.9398

Class 36: 0.9236

Test Mean F1 Score: 0.9164

Test Precision (macro): 0.9166

Test Recall (macro): 0.9165

Train F1 Score per Class:

Class 1: 1.0000

Class 2: 1.0000

Class 3: 1.0000

Class 4: 1.0000

Class 5: 1.0000

Class 6: 1.0000

Class 7: 1.0000

Class 8: 1.0000

Class 9: 1.0000

Class 10: 1.0000

Class 11: 1.0000

Class 12: 1.0000

Class 13: 1.0000

Class 14: 1.0000

Class 15: 1.0000

Class 16: 1.0000

Class 17: 1.0000

Class 18: 1.0000

Class 19: 1.0000

Class 20: 1.0000

Class 21: 1.0000

Class 22: 1.0000

Class 23: 1.0000

Class 24: 1.0000

Class 25: 1.0000

Class 26: 1.0000

Class 27: 1.0000

Class 28: 1.0000

Class 29: 1.0000

Class 30: 1.0000

Class 31: 1.0000

Class 32: 1.0000

Class 33: 1.0000

Class 34: 1.0000

Class 35: 1.0000

Class 36: 1.0000

Train Mean F1 Score: 1.0000

Train Precision (macro): 1.0000

Train Recall (macro): 1.0000

========= Training MLP with Hidden Layers: [512, 256, 128] =========

Train Accuracy: 99.40%

Test Accuracy: 89.72%

Test F1 Score per Class:

Class 1: 0.9402

Class 2: 0.8834

Class 3: 0.9283

Class 4: 0.8460

Class 5: 0.8628

Class 6: 0.9406

Class 7: 0.8947

Class 8: 0.8982

Class 9: 0.9467

Class 10: 0.9338

Class 11: 0.9548

Class 12: 0.9305

Class 13: 0.8878

Class 14: 0.9204

Class 15: 0.9263

Class 16: 0.9542

Class 17: 0.8428

Class 18: 0.8779

Class 19: 0.8532

Class 20: 0.9025

Class 21: 0.8983

Class 22: 0.9433

Class 23: 0.8617

Class 24: 0.8178

Class 25: 0.8605

Class 26: 0.8644

Class 27: 0.9574

Class 28: 0.9371

Class 29: 0.8534

Class 30: 0.9018

Class 31: 0.8885

Class 32: 0.8571

Class 33: 0.8550

Class 34: 0.8742

Class 35: 0.9007

Class 36: 0.9131

Test Mean F1 Score: 0.8975

Test Precision (macro): 0.9004

Test Recall (macro): 0.8972

Train F1 Score per Class:

Class 1: 0.9967

Class 2: 0.9967

Class 3: 0.9983

Class 4: 0.9975

Class 5: 0.9908

Class 6: 0.9992

Class 7: 0.9831

Class 8: 0.9917

Class 9: 1.0000

Class 10: 0.9983

Class 11: 0.9983

Class 12: 0.9950

Class 13: 0.9941

Class 14: 0.9924

Class 15: 1.0000

Class 16: 0.9983

Class 17: 0.9967

Class 18: 0.9856

Class 19: 0.9983

Class 20: 0.9950

Class 21: 0.9975

Class 22: 0.9958

Class 23: 0.9950

Class 24: 0.9804

Class 25: 0.9899

Class 26: 0.9899

Class 27: 1.0000

Class 28: 0.9933

Class 29: 0.9876

Class 30: 0.9975

Class 31: 0.9975

Class 32: 0.9958

Class 33: 0.9693

Class 34: 0.9942

Class 35: 0.9983

Class 36: 0.9975

Train Mean F1 Score: 0.9940

Train Precision (macro): 0.9942

Train Recall (macro): 0.9940

========= Training MLP with Hidden Layers: [512, 256, 128, 64] =========

Train Accuracy: 97.96%

Test Accuracy: 88.38%

Test F1 Score per Class:

Class 1: 0.9410

Class 2: 0.8986

Class 3: 0.9194

Class 4: 0.7816

Class 5: 0.8494

Class 6: 0.8702

Class 7: 0.8790

Class 8: 0.9045

Class 9: 0.9199

Class 10: 0.9475

Class 11: 0.9636

Class 12: 0.9344

Class 13: 0.8694

Class 14: 0.9148

Class 15: 0.9241

Class 16: 0.9508

Class 17: 0.7849

Class 18: 0.8590

Class 19: 0.8107

Class 20: 0.8956

Class 21: 0.7833

Class 22: 0.9556

Class 23: 0.8444

Class 24: 0.8503

Class 25: 0.8449

Class 26: 0.8226

Class 27: 0.9368

Class 28: 0.9347

Class 29: 0.8148

Class 30: 0.9216

Class 31: 0.8804

Class 32: 0.8373

Class 33: 0.8617

Class 34: 0.8795

Class 35: 0.8938

Class 36: 0.9109

Test Mean F1 Score: 0.8831

Test Precision (macro): 0.8871

Test Recall (macro): 0.8838

Train F1 Score per Class:

Class 1: 0.9983

Class 2: 0.9925

Class 3: 0.9756

Class 4: 0.9431

Class 5: 0.9874

Class 6: 0.9561

Class 7: 0.9801

Class 8: 0.9975

Class 9: 0.9877

Class 10: 0.9950

Class 11: 0.9950

Class 12: 0.9860

Class 13: 0.9827

Class 14: 0.9876

Class 15: 0.9857

Class 16: 0.9933

Class 17: 0.9621

Class 18: 0.9738

Class 19: 0.9557

Class 20: 0.9908

Class 21: 0.8580

Class 22: 0.9899

Class 23: 0.9834

Class 24: 0.9882

Class 25: 0.9835

Class 26: 0.9715

Class 27: 0.9901

Class 28: 0.9992

Class 29: 0.9521

Class 30: 0.9967

Class 31: 0.9744

Class 32: 0.9753

Class 33: 0.9901

Class 34: 0.9925

Class 35: 0.9925

Class 36: 0.9892

Train Mean F1 Score: 0.9792

Train Precision (macro): 0.9804

Train Recall (macro): 0.9796

========== AVERAGE PERFORMANCE ACROSS ALL MLP MODELS ==========

Average Train Accuracy: 99.34%

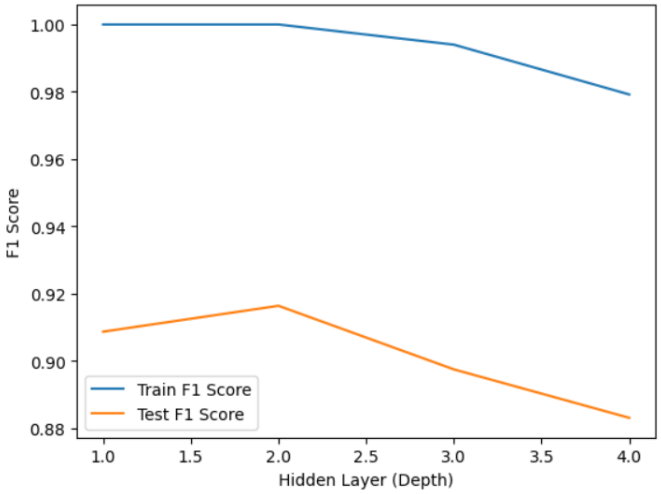
Average Test Accuracy: 90.16%

Average Test F1 Score: 0.9014

Average Test Precision: 0.9033

Average Test Recall: 0.9016

Here 1 = [512] , 2 =[512,256] ……. 4 = [512,256,128,64]

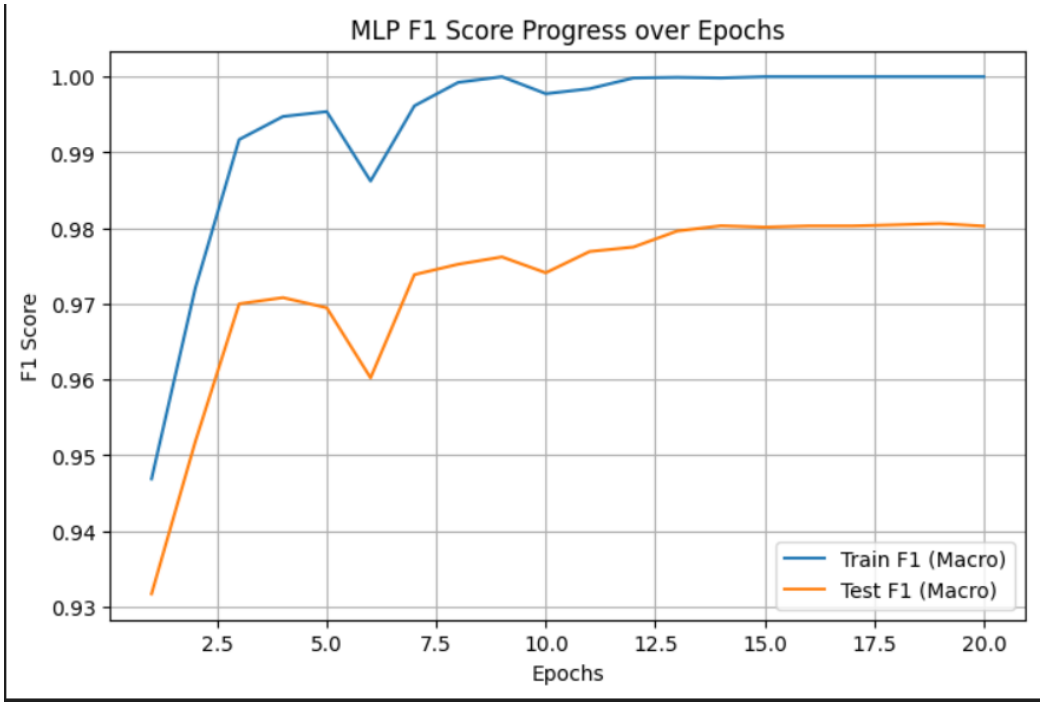


Comments – Approximately same average accuracies for both the network in part d and e , indicating similar model trained by me and implementation by the Sk- learn MLP classifier.Sk-Learn seems to be slightly better.Stopping crietria was max epochs of 50.

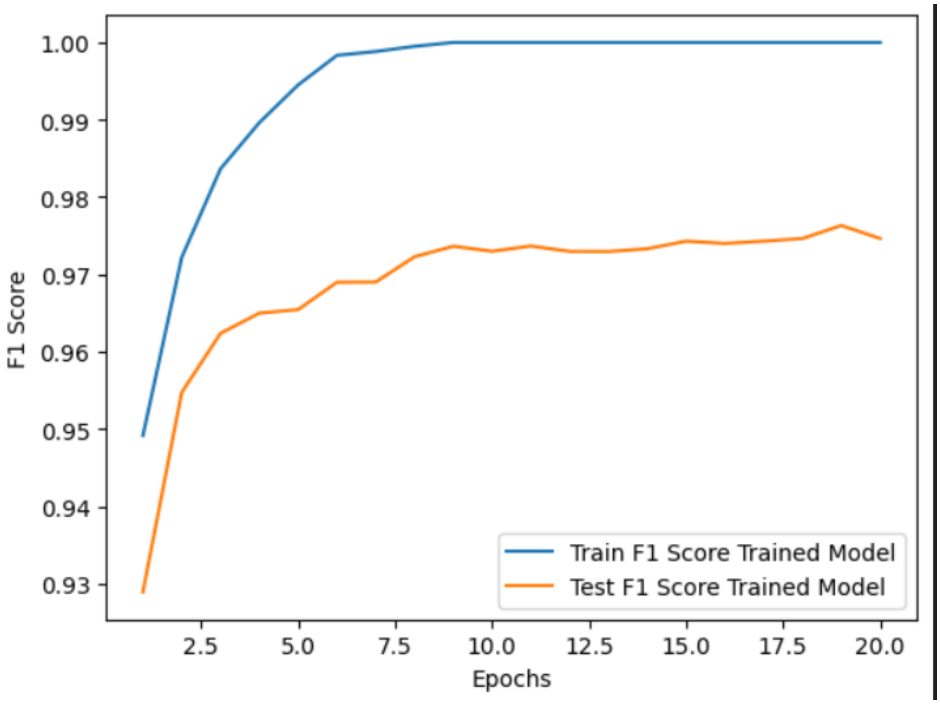
Architecture	Self	Sk-learn
512	86.19	90.88
512,256	87.4	91.65
512,256,128	87.6	89.72
5512,256,128,64	88.9	88.38
Average	~87	90.16

(F)

(A)


MLP F1 Score Progress over Epochs
F1 Score
Epochs
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Train F1 (Macro)
Test F1 (Macro)

(B)


F1 Score
Epochs
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Train F1 Score Trained Model
Test F1 Score Trained Model

Comments -

The results demonstrate that transfer learning significantly improves both convergence speed and generalization performance. When trained from scratch on the digits dataset, the model (Figure A) shows a slower rise in F1-scores during the initial epochs. In contrast, the fine-tuned model initialized with the pre-trained weights from the consonant dataset (Figure B) achieves a higher test F1-score more rapidly, indicating that the pre-learned feature representations effectively transfer to the digit classification task. This improvement arises because the earlier layers capture generic visual features such as strokes and shapes that are common across consonants and digits. Also transfer learning model , overfits quickly and becomes stagnant to further epochs.