1.

a.

experiment	classes	n	hidden layer	max iter	Us	sklearn	logreg
1	2,5	10000	8	200	0.97	0.97	0.976
2	1,5	10000	8	200	0.98	0.997	0.995
3	0,1	1000	8	200	1	1	1
4	0,1	5000	8	200	0.995	1	1
5	0,1	9000	8	200	1	0.997	0.997
6	0,1	8000	8	200	0.97	0.97	0.976
7	0,1	15000	8	200	0.995	0.997	0.998
8	0,1	20000	8	200	0.994	0.998	0.999
9	0,1	10000	15	200	0.995	1	1
10	0,1	10000	10	200	0.991	0.993	1
11	0,1	10000	9	200	0.998	0.993	1
12	0,1	10000	7	200	0.995	0.993	1
13	0,1	10000	6	200	0.993	0.995	1
14	0,1	10000	5	200	0.984	0.979	1
15	0,1	10000	4	200	0.993	0.993	1
16	0,1	10000	3	200	0.995	0.988	1
17	0,1	10000	2	200	0.998	0.998	1
18	0,1	10000	1	200	0.993	0.993	1
19	0,1	10000	8	250	0.993	0.998	1
20	0,1	10000	8	150	0.993	0.998	1
21	0,1	10000	8	100	0.993	0.998	1
22	0,1	10000	8	50	0.991	0.998	1
23	0,1	10000	8	20	0.986	0.998	0.995
24	0,1	10000	8	10	0.981	0.998	0.993
25	0,1	10000	8	5	0.948	0.998	0.995

For class, we expect numbers that are similar to each other to have worse result than numbers that are similar. The more similar the number, the more likely the algorithm will mistake the two numbers. Like in experiment 2 and 3, I use number 5 and 2 as similar classification numbers and 5 and 1 as unsimilar classification numbers. The results, albeit high, match, with experiment 2 having a lower result across all algorithms. We expect the results to increase as max iteration increases up to a certain point. The higher max iteration is, the more chances the model has to converge. However, if the model converges below the max iteration, increasing the max iteration should have no affect. This is seen in experiment 19 to 25. Experiment 19 to 21, experiments with 100 plus iterations, all have the same high results while after experiment 21 the result start to decrease.
Increasing N should increase the results. Since, N correlates to the number of examples the model trains on, the more examples the models can train, the better the algorithm

is fitted and therefore the better the result. Experiment 3 to 8 do not seem to fit this trend because with N = 1000 the model has the highest result while all the other experiments with higher N have lower scores.
Results should increase when the hidden layer increases. With more hidden layers, there are more filters/weights to distinguish the classifications. Experiment 9 to 18 doesn't really show this and there isn't really a clear trend. However, I did only test on two very distinguishable class values (0 and 1) so that could be why some of the results are not what expected because the results themselves are all still very high even though they do not match the trend.

b.

experiment	classes	n	hidden layer	max iter	sklearn	logreg
1	range 10	10000	32	100	0.916	0.876
2	range1,10	10000	32	100	0.878	0.876
3	range2,10	10000	32	100	0.881	0.854
4	range3,10	10000	32	100	0.878	0.866
5	range4,10	10000	32	100	0.885	0.907
6	range5,10	10000	32	100	0.921	0.909
7	range6,10	10000	32	100	0.938	0.946
8	range7,10	10000	32	100	0.928	0.922
9	range8,10	10000	32	100	0.974	0.971
10	range 10	15000	32	100	0.912	0.893
11	range 10	9000	32	100	0.903	0.883
12	range 10	8000	32	100	0.891	0.872
13	range 10	4000	32	100	0.853	0.892
14	range 10	10000	64	100	0.892	0.895
15	range 10	10000	16	100	0.871	0.895
16	range 10	10000	8	100	0.107	0.895
17	range 10	10000	32	200	0.912	0.886
18	range 10	10000	32	50	0.876	0.913
19	range 10	10000	32	10	0.652	0.896
20	(1,0)	15000	64	200	0.995	0.998

As the number of classification decreased, the higher the results of the algorithm became. The experiments also showed that decreasing n, hidden layer, and max iteration, decreased the results which aligns with what was said in part a. The best setting was experiment _ settings because it has the most distinguishable pair of classes, and high n, hidden layer, and max iteration.

c. The reason the why our current model cannot work in ten_class is because our losses only work with binary classes (aka two classes). To be able to work with ten_class, we need to incorporate a loss function that can have multiple classes.