COMS 574 Final Project Report

Justin Stanley

December 9, 2022

1 Introduction

For this project I conducted experiments analyzing the behavior of different classification models, including SVMs, decision trees, random forests, and neural networks.

2 Experiments (Part 1)

2.1 Cross-validation by method

I used K-fold (K=5) and LOOCV cross validation to compare each classification method on each dataset. 100 trees are used for the random forest. The random forest has the strongest test accuracy on dataset 1, however took the longest time to train. This was especially long when LOOCV cross-validation is used, as the training time is quadratic in the number of samples. All classifiers performed significantly worse than dataset 1, suggesting dataset 2 contains more evenly distributed/mixed features. The random forest exhibited the strongest performance across each dataset. The LOOCV cross-validation produced generally similar results to K-fold while taking much longer to compute.

The results are formatted by the following variables:

Classifier	C
Validation method	V
Train accuracy	A_T
Train precision	P_T
Train recall	R_T
Train F1	F_T
Test accuracy	A_S
Test precision	P_S
Test recall	R_S
Test F1	F_S

Dataset 1

C	**V**	A_T	A_S	P_T	P_S	R_T	R_S	F_T	F_S
SVM	K-fold	0.92	0.916	0.968	0.972	0.816	0.797	0.442	0.434
SVM	LOOCV	0.922	0.912	0.966	0.955	0.819	0.802	0.443	0.5
DecisionTree	K-fold	1	0.917	1	0.876	1	0.904	0.5	0.444
DecisionTree	LOOCV	1	0.921	1	0.885	1	0.906	0.5	0.5
RandomForest	K-fold	1	0.963	1	0.965	1	0.933	0.5	0.474
RandomForest	LOOCV	1	0.963	1	0.957	1	0.943	0.5	0.5

Dataset 2

C	**V**	A_T	A_S	P_T	P_S	R_T	R_S	F_T	F_S
SVM	K-fold	0.675	0.660	0.73	0.708	0.094	0.076	0.098	0.099
SVM	LOOCV	0.668	0.662	0.721	0.643	0.065	0.056	0.06	0.5
DecisionTree	K-fold	1	0.593	1	0.427	1	0.459	0.5	0.219
DecisionTree	LOOCV	1	0.584	1	0.405	1	0.425	0.5	0.5
RandomForest	K-fold	1	0.662	1	0.531	1	0.395	0.5	0.222
RandomForest	LOOCV	1	0.697	1	0.583	1	0.438	0.5	0.5

For simplification, the Precision, Recall, and F1 data are omitted from the remainder of the experiments, as they are not significantly related to the analysis of overfitting and underfitting.

2.2 SVM Regularization

For regularization experiments I manipulated the parameter C used by the scikit-learn SVC. I tested the values $C = 0.1, 1, 10, 50, 100, 1000$. I found that as the regularization constant increased, the SVC had better training and testing accuracy, however the model took slightly longer to train.

Dataset 1

C	Accuracy (train)	Accuracy (test)
0.1	0.893	0.889
1	.920	.919
10	.923	.925
50	.933	.933
100	.941	.931
1000	.956	.958

Dataset 2

C	Accuracy (train)	Accuracy (test)
0.1	0.654	0.654
1	0.673	0.664
10	0.733	.699
50	0.756	0.695
100	.766	.719
1000	.810	0.697

2.3 Decision tree splitting

I analyzed the performance of GINI, entropy and log-loss methods for training decision trees and random forests. The random forest using entropy-based splitting performed best on dataset 1, while the random forest using either GINI or log-loss performed best on dataset 2.

Dataset 1

Method	Split	Accuracy (train)	Accuracy (test)
DecisionTree	GINI	1	.940
DecisionTree	Log-loss	1	.897
DecisionTree	Entropy	1	.939
RandomForest	GINI	1	.956
RandomForest	Log-loss	1	.96
RandomForest	Entropy	1	.963

Dataset 2

Method	Split	Accuracy (train)	Accuracy (test)
DecisionTree	GINI	1	.602
DecisionTree	Log-loss	1	.597
DecisionTree	Entropy	1	.621
RandomForest	GINI	1	.686
RandomForest	Log-loss	1	.686
RandomForest	Entropy	1	.673

2.4 Random forest tree count

For this experiment I evaluated the performance of random forests with 10, 100, 500, 1000, and 10000 trees. The testing accuracy generally increased as the number of trees increased, as expected. The training time was significantly longer for the forests with 10000 trees.

Dataset 1

Tree count	Accuracy (train)	Accuracy (test)
10	0.999	0.958
100	1	0.961
500	1	0.958
1000	1	0.956
10000	1	0.965

Dataset 2

Tree count	Accuracy (train)	Accuracy (test)
10	0.979	0.651
100	1	0.677
500	1	0.677
1000	1	0.68
10000	1	0.68

2.5 Summary

By the experiments performed above, the strongest classifier for dataset 1 is a random forest training on entropy loss (test acc 0.963), with 10000 trees (test acc 0.965). The strongest classifier for dataset 2 is an SVM with regularization constant $C = 100$ with a test accuracy of 0.719. As expected, plain decision trees exhibited strong overfitting (100% train accuracy and ≈0.6 test accuracy), and are likely not a good choice for either dataset.

3 Experiments (Part 2)

3.1 Hidden units

For this experiment I evaluated the performance of the model with 12, 32, 64, and 128 units. The results are visualized in figure 3.1. The results show the model accuracy converges fastest when 12 hidden units are used.

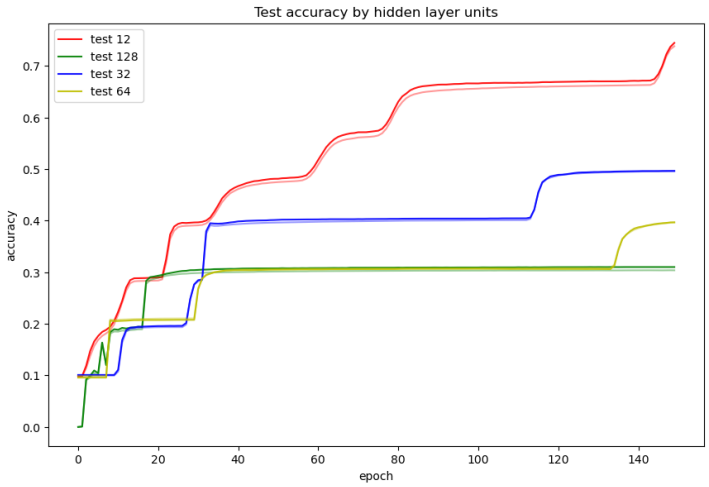

Figure 1: Network accuracy over time by hidden units

3.2 Learning rate

For this experiment I evaluated the performance of the model with learning rates of 0.001, 0.005, and 0.01. The results are visualized in figure 3.2. The results show the model accuracy converges fastest with a learning rate of 0.005, but finishes with the best accuracy at learning rate 0.005.

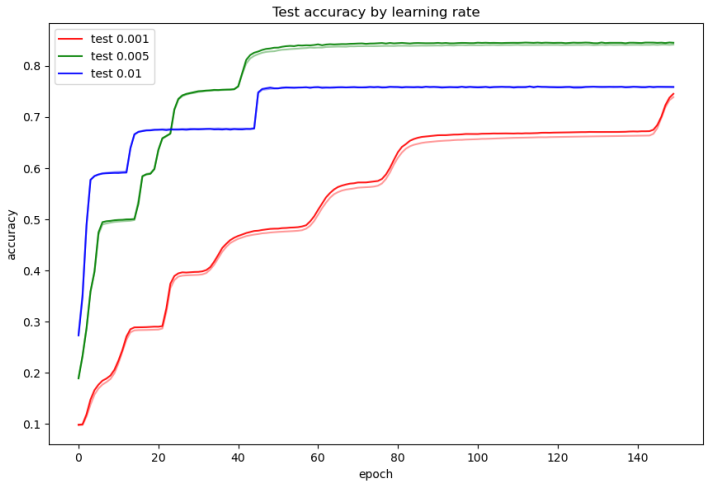

Figure 2: Network accuracy over time by learning rate

3.3 Regularization

For this experiment I evaluated the performance of the model with different weight regularization. I tested L2 regularization multipliers of 0, 0.001, and 0.003. The results are visualized in figure 3.3. The results show the model accuracy converges fastest with no regularization, and performs generally worse as regularization increases.

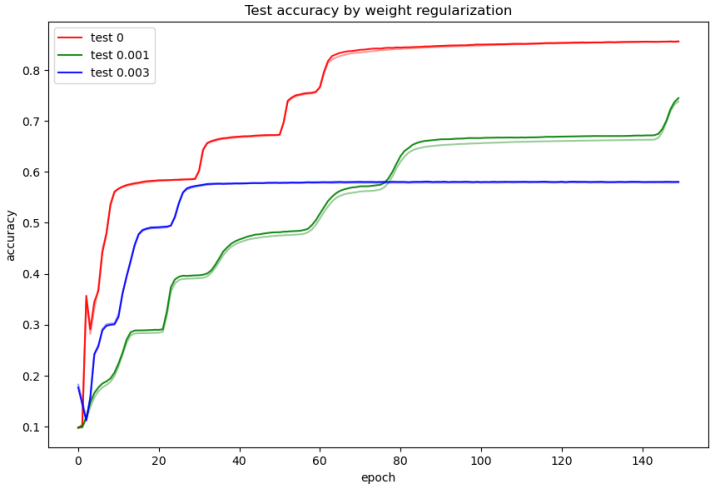

Figure 3: Network accuracy over time by L2 regularization weight

3.4 Activation

For this experiment I evaluated the performance of the model with different activation functions. I tested the model using relu, sigmoid, and tanh activation. The results are visualized in figure 3.4. The results show the model accuracy converges fastest with tanh activation, and slowest with sigmoid.

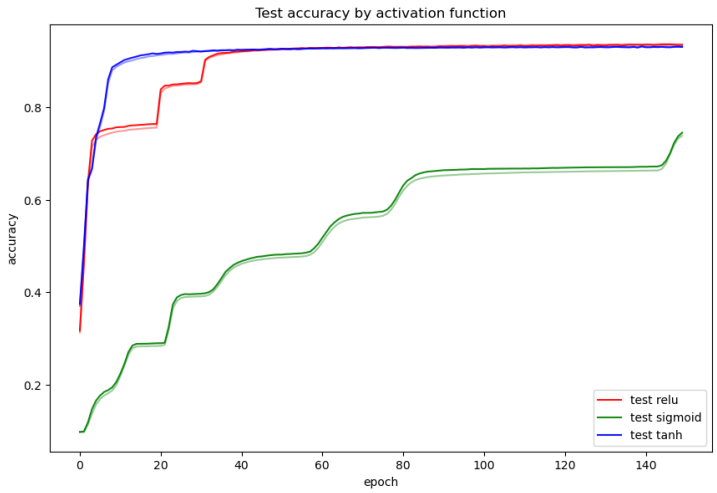

Figure 4: Network accuracy over time by activation function

3.5 Summary

By the experiments performed above, the final best model is one with 12 hidden units by 2 layers, tanh activation function, with learning rate 0.005 and no regularization. The training accuracy generally closely followed the training accuracy, exhibiting minimal overfitting with a maximum difference of about 2%.

The model did exhibit underfitting with a small learning rate as shown above with $\gamma = 0.001$.

Experimental data for part 2 is available in the directory `part2/experiments/`.

4 Source code

The code for parts 1 and 2 of project are provided in the `/part1` and `/part2` directories respectively. The usage information for each program can be displayed with the following commands:

```
$ python part1.py --help
$ python part2.py --help
```