CMPUT 466 Project: Cancer Prediction

Introduction:

In this project, I will formulate the Cancer prediction problem as a Binary Classification (either malign or benign) machine learning problem. This data sample is small and got from Kaggle. In order to classify between malign and benign, we use different features including radius_mean, texture_mean, perimeter_mean, and other total 30 features. I will use train-validation-test framework and predict 1 for malign and 0 for benign.

Problem formulation:

There are 30 features in the database for each 569 input data samples. Output is "diagnosis", either benign(0) and malign (1).
Split sample as 3 set, one for training (400), one for validaeiton test (100) and one for test (69).

Approach and Baseline:

Algorithm1 : Logistic Regression

Max_epoch = 1000
Baseline: stepsize = 0.1
Hyperparameter: stepsize parameter
How to tune the parameter:
Set parameter from range [0.1, 0.001, 0.0001]

Algorithm 2: neural network

Max_epoch = 1000
baseline: nn model1
Baseline accuracy: 0.6666666666666666
Hyperparameter: number of layers and nodes in each layer
How to tune the parameter:
Train Model1 using 3 layers with nodes 15, 10, 1 respectively
Train Model 2 using 2 layers with 4, 1 respectively

Algorithm3: Decision Tree

Baseline: depth with 1
Hyperparameter: max tree depth
Tuing the hyperparameter: range from [1,10]

Evaluation metric

Usig the measure of success(accuracy) in Dtest to evaluate. This is not goal of the task but an reasonable appraoximation. Since the test sample is randomly selected and could represent the true problem in some scales.

Algorithm1 : Logistic Regression

Measure of success: Accuracy on Test set
Best epoch

Algorithm 2: Neural network:

Measure of success: Accuracy on test set
Confusion matrix
Precision_score
recall score
f1 score

Algorithm 3: Decision Tree

Accuracy in Test set

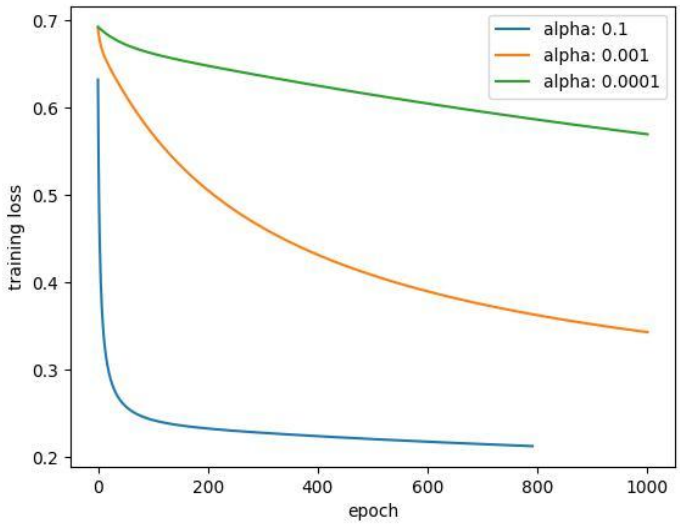
Result

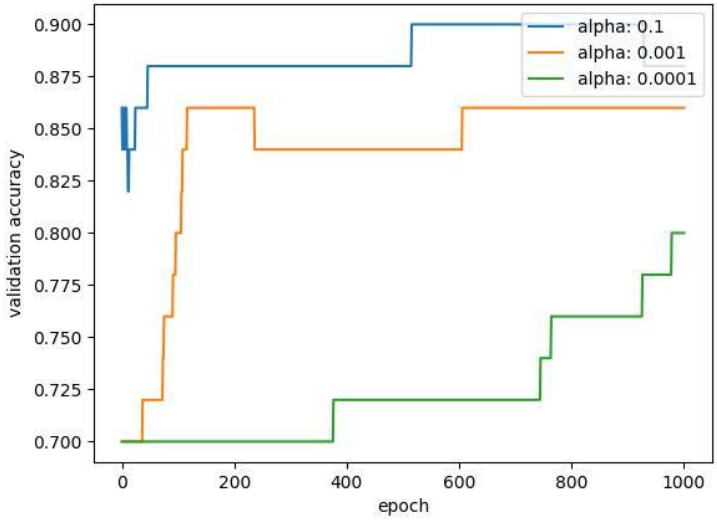
Algorithm1 : Logistic Regression

Best epoch: 999

Accuracy of Logistic regression on D_test: 0.971014492753623

Change alpha from 0.1 to less value will decrease the speed of learning, and with only 1000 epoch, the training loss and validation accuracy will not decrease as quick as possible.





Algorithm 2: neural network

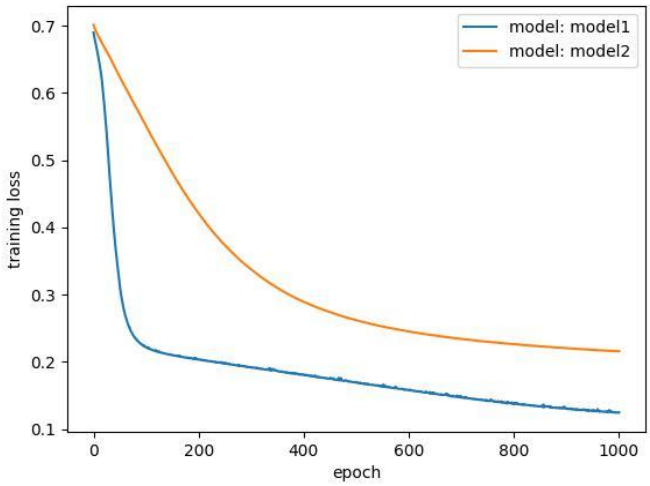
Model1 (with more layers and nodes in each layer)

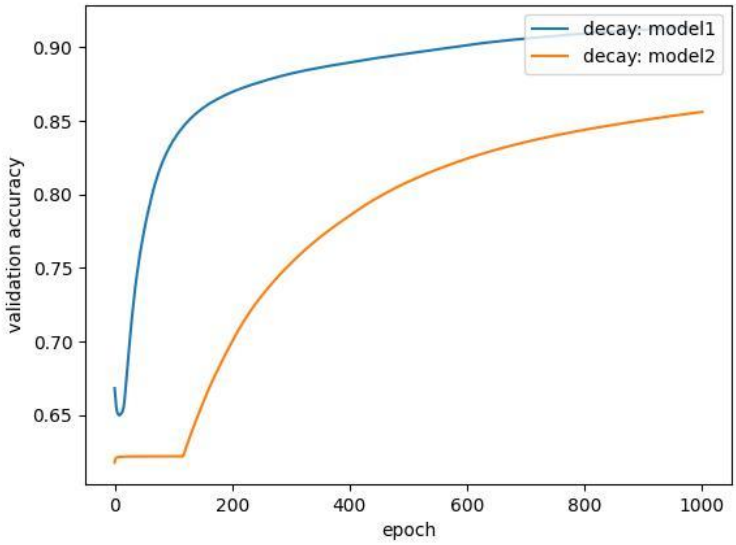
Confusion matrix for nn model1:
[[46 0]
[2 21]]
precision_score: 1.0
recall score: 0.9130434782608695
f1 score: 0.9545454545454545
Messure of Success : 0.9150100946426392

Model2 (less layers)

Measure of Success: 0.856082558631897 (worse)

Model 1 is much better than model as the training loss decrease faster and the validation accuracy is better than model 2. In addition, the precision score, recall score and f1 score is pretty high, training is good. Besides, the test accuracy in model 1 is better than model 2, and it's about 92% and has a good performance in the testing samples.

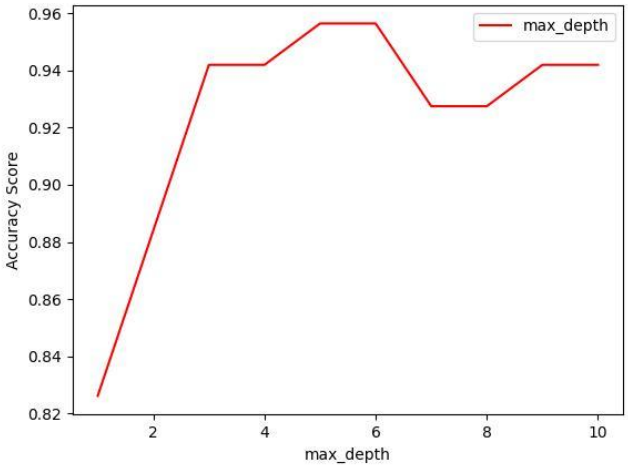




Algorithm3: Decision Tree

Compared to baseline max depth = 1, the accuracy increase before reaching depth7, with Max depth = 5 or 6 with accuracy in D_Test = 0.9565217391304348. And then accuracy decrease.

Increaseing depth to larget than 6 will cause overfitting and decrease test accuracy


max_depth
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.82
Accuracy Score
2
4
6
8
10
max_depth