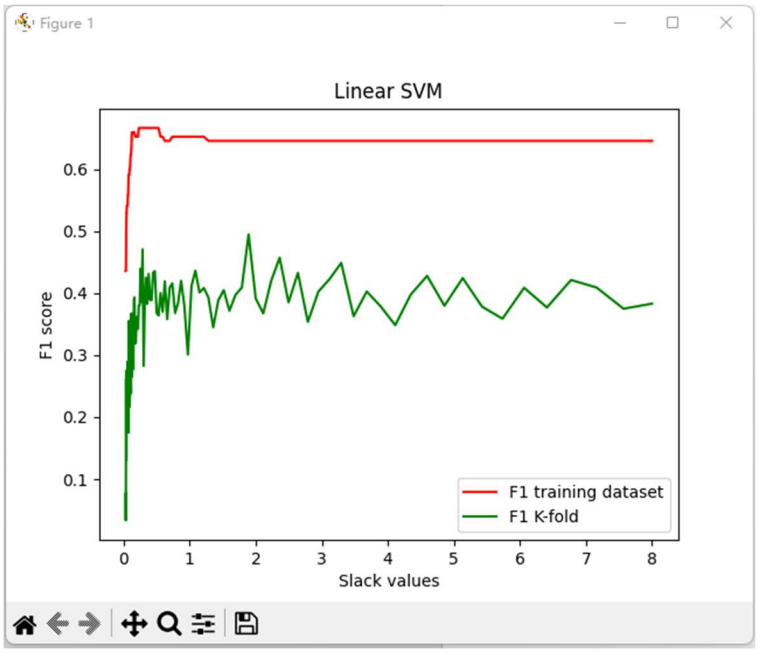
CSC 246 Project2
Junfei Liu(jliu137)

Stage One – SVMs

1. LinearSVM with various slack variable parameter



Maximum numerical observation:

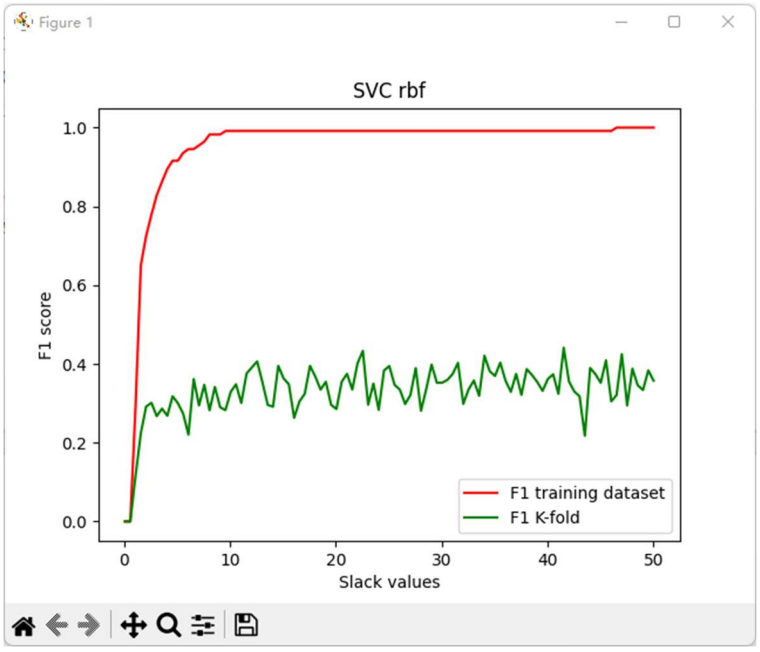
F1 score on training set: 0.6666666666666666 with parameter 0.23004691265621877

accuracy score on training set: 0.8306878306878307 with parameter 0.125

F1 score on cross validation: 0.4766666666666667 with parameter 0.33915108186191795

accuracy score on cross validation: 0.7304409672830725 with parameter 0.33915108186191795

2. SVC rbf kernel with various slack variable parameter



Maximum numerical observation:

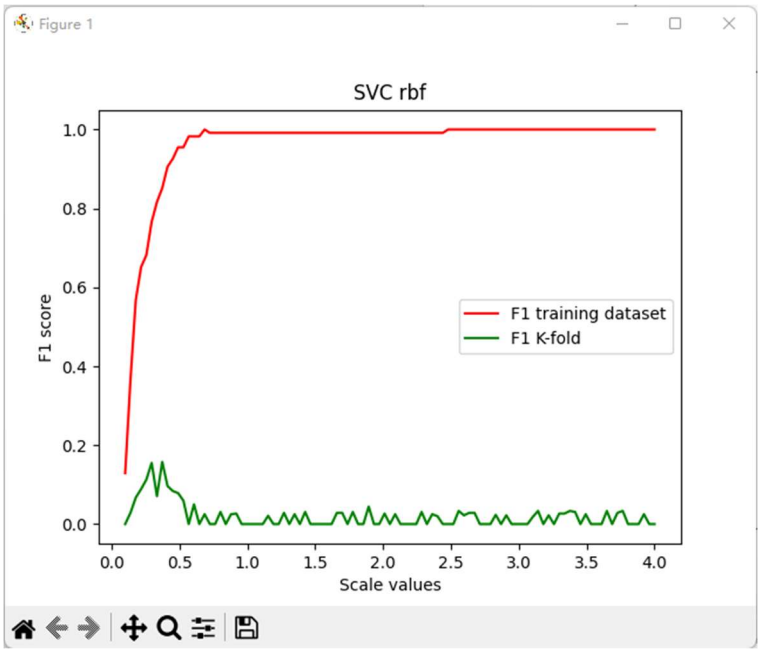
F1 score on training set: 1.0 with parameter 46.507

accuracy score on training set: 1.0 with parameter 46.507

F1 score on cross validation: 0.44105627705627704 with parameter 41.517

accuracy score on cross validation: 0.7086770981507823 with parameter 1.098

3. SVC rbf kernel with various scale variable parameter



Maximum numerical observation:

F1 score on training set: 1.0 with parameter 0.6849999999999999

accuracy score on training set: 1.0 with parameter 0.6849999999999999

F1 score on cross validation: 0.15762032085561498 with parameter 0.373

accuracy score on cross validation: 0.7042674253200569 with parameter 0.373

4. Evaluation & Reflection

The LinearSVC model can not reach 100% accuracy and F1 score while the SVC with rbf kernel can. The training time generally increase with larger slack parameters and does not vary significantly with different scale parameters. However, SVM is trained fast enough that human cannot really realize the difference.

Stage Two – MLPs

1. Performance with 10 hidden units

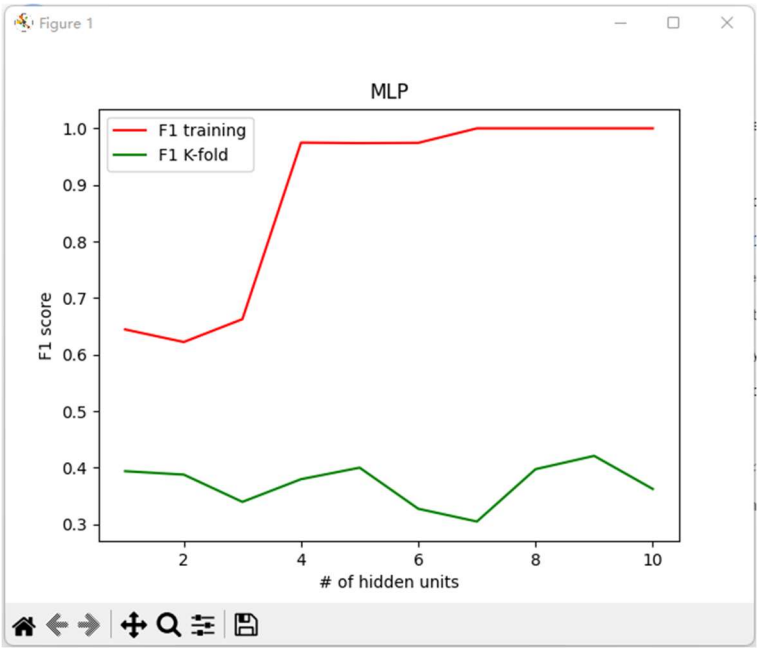
Minimum loss is 0.03537740558385849 and accuracy is 1.0 with learning rate of 0.20

In which loss is keras.losses.BinaryCrossentropy() with optimizer="SGD"(stochastic gradient descent). It runs with min-max scaled data, 10 hidden units, 200 epochs, and learning rate of 0.20.

The annotated code in function train_model and ten_hidden in nn.py shows that a learning rate of 0.20 can reach 100% accuracy by iterating the learning rate from 0.01 to 1 and 0.20 performs the best.

Because a network with 10 hidden units can already reach perfect memorization, I will skip question#2.

3. Sweeping the number of hidden units from 1 to 10



The models are also trained with optimizer="SGD", min-max scaled data, 10 hidden units, 200 epochs, and learning rate of 0.20.

Summary

I prefer kernel SVM because

1. SVM with rbf kernel is trained significantly faster than MLP while they both can reach 100% accuracy on training data and around 40% on cross-validation;
2. linearSVM fails to reach 100% accuracy on training data while training times are similar.

Therefore, kernel SVM is generally superior in terms of performance and accuracy.

Run Instruction

You can run and recreate my results for SVM and MLP by running svm.py and nn.py respectively. They will execute and show the results corresponding to the requirements in the same order as in the assignment writeup.

I am working solo with no partner.