COMP4641 Assignment #1
Liu Qinhan (qliu359@gatech.edu)

Supervised Learning Report

Dataset

Breast Cancer Data This is a set of data about the breast cancer diagnosis, that is, whether the tested cell is of malignant (bad) or benign (mild). 357 benign and 212 malignant samples with 30 features of each sample are included. The features are computed, as described on the website, based on the digitized images of a fine needle aspirate (FNA) of a breast mass. In the training process, the classification is indicated by "1" (malignant) and "0" (benign).

Credit Card Fraud Data This dataset contains totally 284,807 transactions made by European cardholders in Sep. 2013. 492 frauds are recorded inside taking up only 0.172% of all transactions. 29 features are included and the fraud transactions are classified by "1".

Why interesting?

A simple answer: both of the datasets are of great practical applications in daily life.

The breast cancer is now the No.2 causes for women's death. According to the *report* from American Cancer Society, a total of 252,710 new cases among women and 2,470 for men of breast cancer were expected to be diagnosed in 2017. Approximately 40,610 women and 460 men were expected to die from breast cancer in 2017. Machine learning models can be of great help to the medical team when diagnosing the cancer because of its massive experiences based on past data. Its high performance in making real time decisions saves people valuable time for a better and more effective healthcare.

About 70% people in the US own credit cards and most of them have more than 1 card. Nobody owns a credit card without paying with it. In fact, as the credit card prevails because of its convenience in modern commerce, new kinds of credit card frauds keep coming out. If the banks equip themselves with this tool, which predicts whether or not the transaction of the customers involves fraud based on the previous transactions, it will be much safer to use a credit card.

In the context of machine learning, the breast cancer data is perfect for training models in may aspects. The data is of such a size that every algorithm can run fast on it. Also, the number of positive and negative examples is about 1:1. In fact, in the analysis, all 5 models are reasonably good to fit the data. The credit card fraud data is instead, a fairly large set, and much more unbalanced with only ~0.1% positive cases. But it is still a good case for applying machine learning into solving real problems.

Decision Trees

Cancer data:

(unpruned)

```
train_size | accu | node# | tree_dep
   426    | 0.916 |  39   |   7
Classification report:
          precision   recall  f1-score   support

   0.0     0.87     0.98     0.92       91
   1.0     0.95     0.75     0.84       52

avg / total     0.90     0.90     0.89      143
```

(pruned)

```
train_size | accu | node# | tree_dep
   426    | 0.944 |  25   |   5
Classification report:
          precision   recall  f1-score   support

   0.0     0.96     0.96     0.96       91
   1.0     0.92     0.92     0.92       52

avg / total     0.94     0.94     0.94      143
```

Card data:

(unpruned)

```
train_size | accu | node# | tree_dep
 213605  | 0.9993 |  303  |   21
Classification report:
          precision   recall  f1-score   support

    0.0     1.00     1.00     1.00   71081
    1.0     0.79     0.81     0.80     121

avg / total     1.00     1.00     1.00    71202
```

(pruned)

```
train_size | accu | node# | tree_dep
 213605  | 0.9995 |  113  |   10
Classification report:
          precision   recall  f1-score   support

    0.0     1.00     1.00     1.00   71081
    1.0     0.92     0.80     0.85     121

avg / total     1.00     1.00     1.00    71202
```

The pruning is done by setting the parameter of the DecisionTreeClassifier: max_depth, which is a pre-pruning method limiting the final tree depth as described *here*. During the training process, several observations are made. First, the pruning increased the model accuracy for cancer data slightly while has a negligible impact on the card data. While, due the pruning strategy, the sizes of the resulting decision trees are reduced notably. I believe that by limiting the depth, the tree stops growing before overfitting for the cancer data, hence the accuracy improvement.

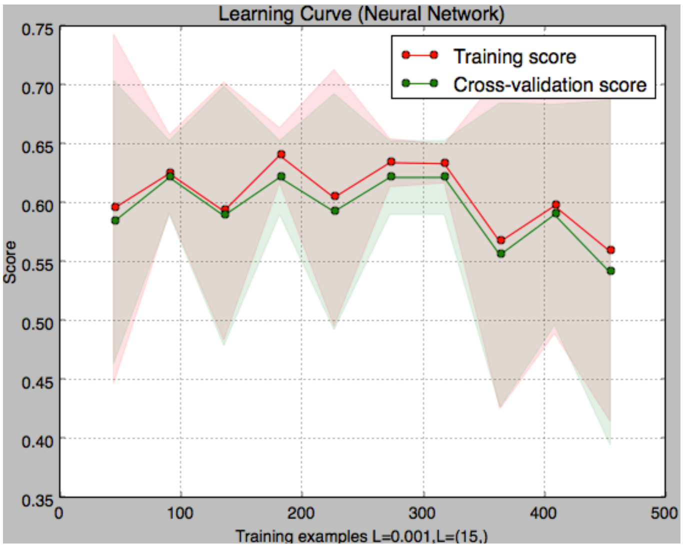
Second, the 2 datasets differ hugely in their size. Following the same training process, card data takes much more time to finish, which is what we would expect intuitively.

Finally, since the card data is highly unbalanced, even 99.95% accuracy still does not mean that we have a good estimation of the problem. Since simply guessing “0” would achieve 98.3% accuracy already. The method to measure our model in such a extreme dataset could be improved. I have been thinking about methods like adding the "weights" to the positive examples by repeating them. But still, they are not real positive cases.

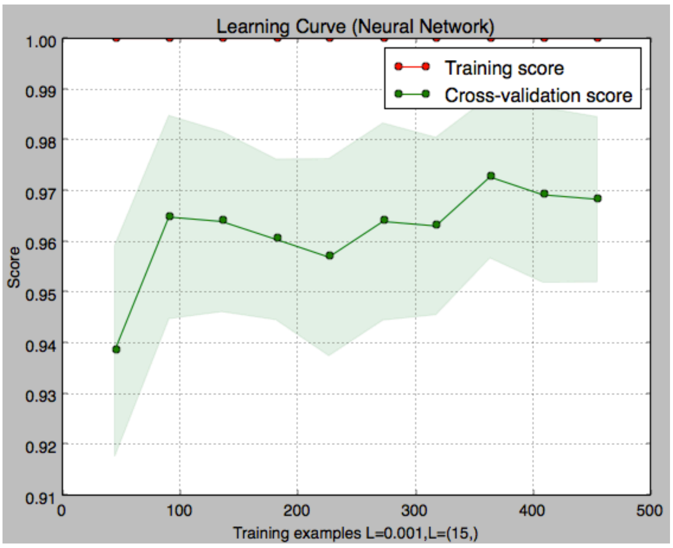
Neural Networks

(1) Data Preprocessing

(Cancer data used)



no preprocessing



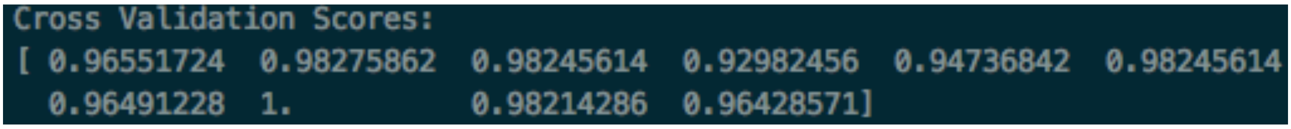
preprocessing

The first issue of neural networks is that this particular model is very sensitive to feature scaling, a single normalization to the raw data could increase the accuracy drastically as if they are 2 models. Thus, in the latter discussions, all data are preprocessed before the fitting.

(2) Cross Validation

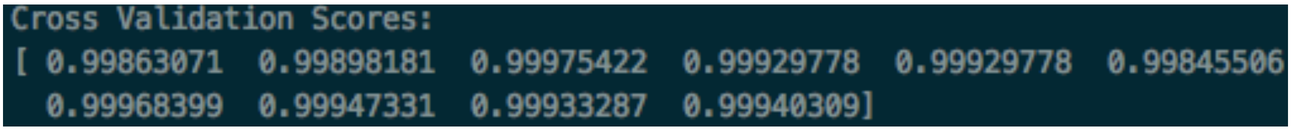
A 10 fold cross validation is performed for both datasets:

Cancer data:



```
Cross Validation Scores:
[ 0.96551724  0.98275862  0.98245614  0.92982456  0.94736842  0.98245614
  0.96491228  1.          0.98214286  0.96428571]
```

Card data:



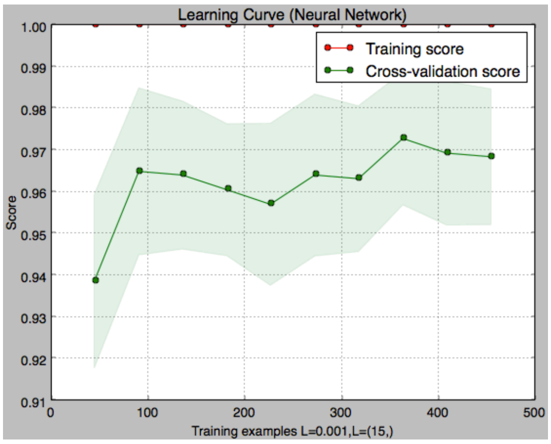
```
Cross Validation Scores:
[ 0.99863071  0.99898181  0.99975422  0.99929778  0.99929778  0.99845506
  0.99968399  0.99947331  0.99933287  0.99940309]
```

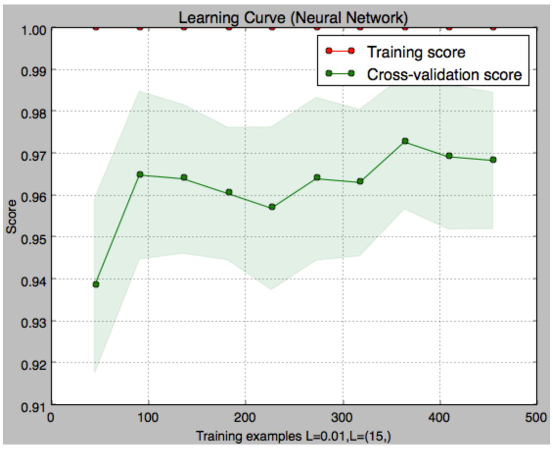
For the cancer data, although there are some differences between folds (0.93-1.0), but they are acceptably high already, so I would say there is no preference in the dataset for the model. For the card data, it is more obvious to derive the same conclusion.

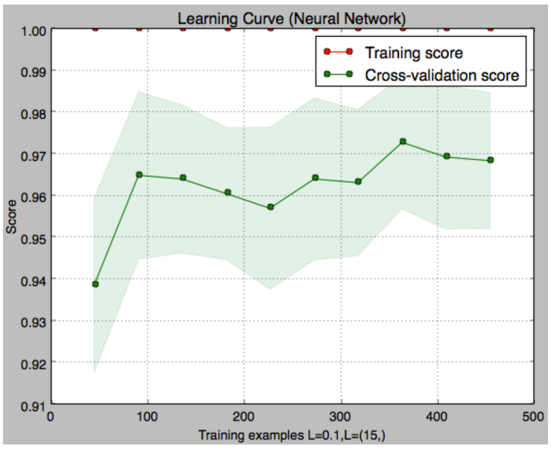
(3) Learning rate analysis

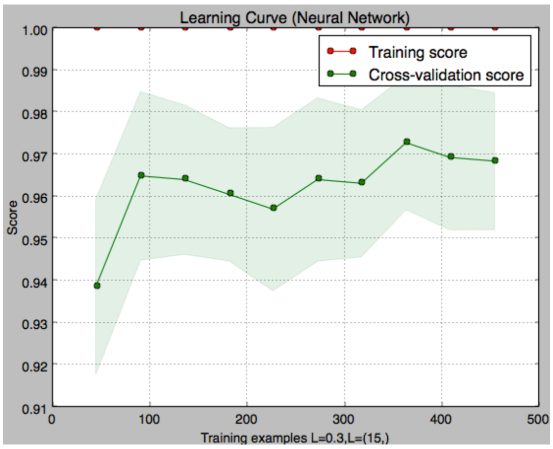
Due to that the size of card data is too large, the same algorithm runs perfectly on it but much more time are needed. In my case, it took me 30min.

Cancer data

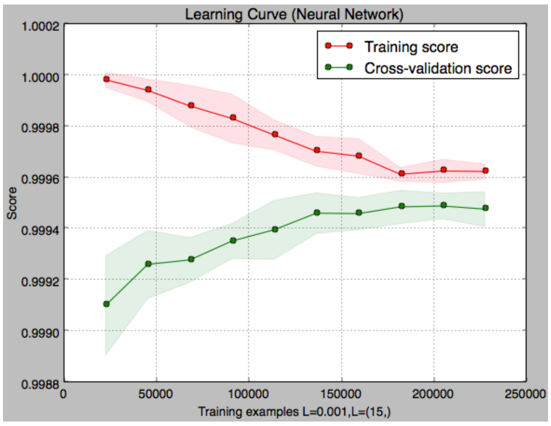


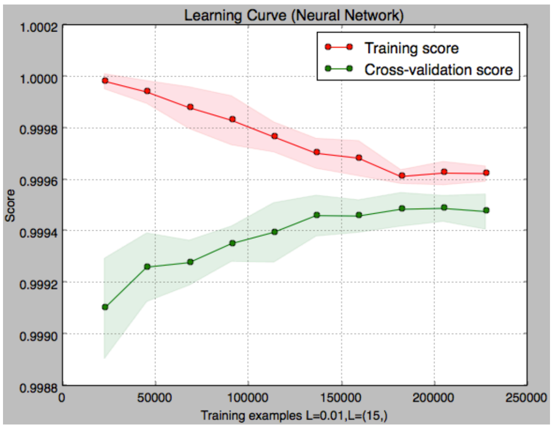


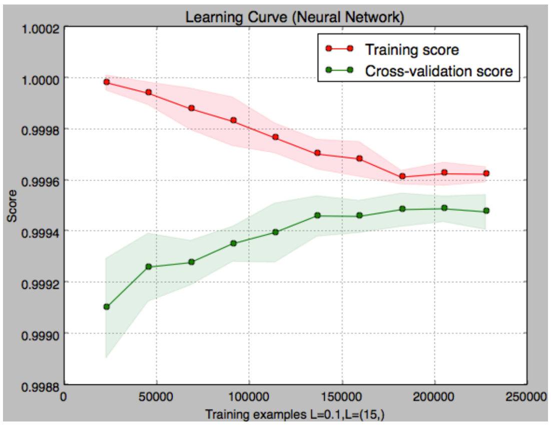


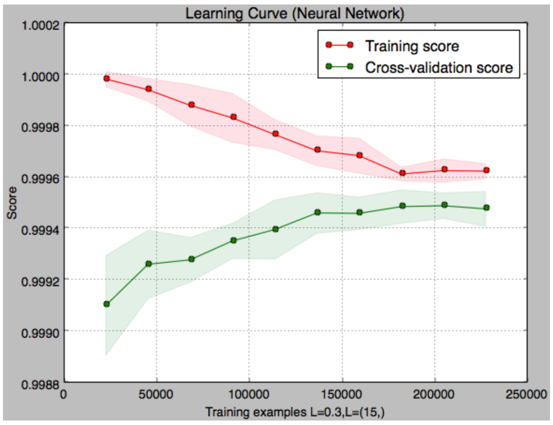


Card data







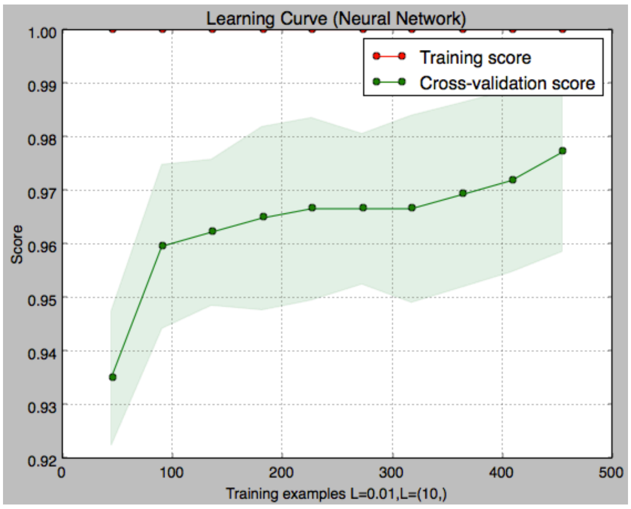


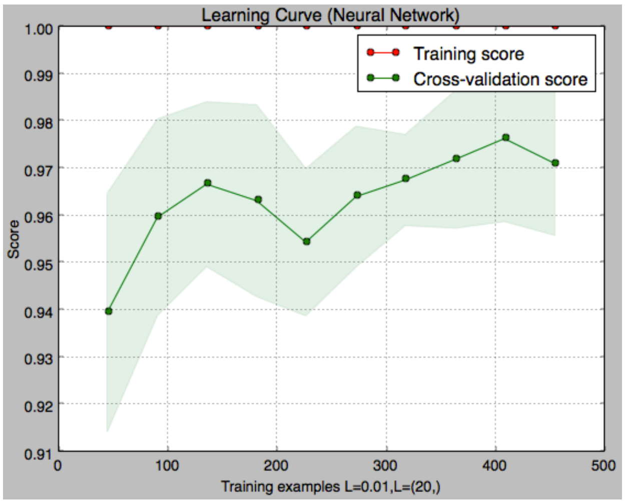
The results does not vary too much as the learning rate do (0.001, 0.01, 0.1, 0.3). I myself am really questionable about this outcome.

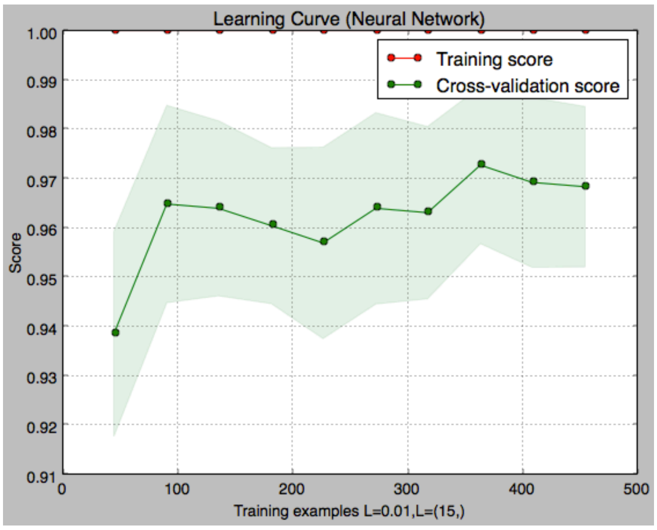
The only one explanation I can come up with is that based on the results of (2), there is no preference in the 2 datasets. This means that regardless which portion is fitted to the model, the resulting model would be pretty similar in general, then the learning accuracy is purely counted on the number of training examples. The reason why these two datasets are like this may due to a randomization process by the data provider.

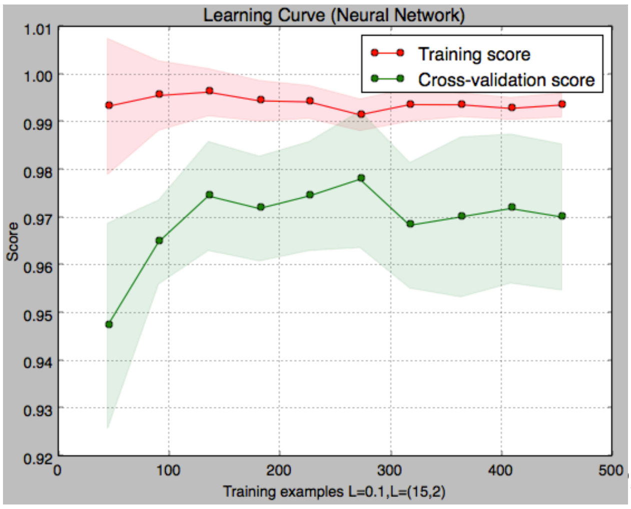
(4) Network architecture analysis

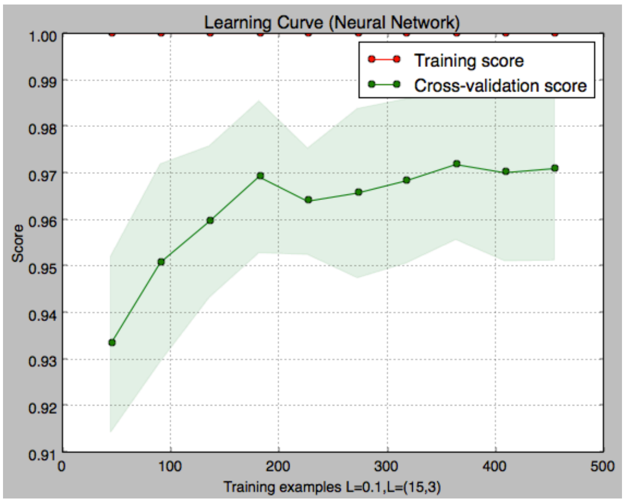
Cancer data



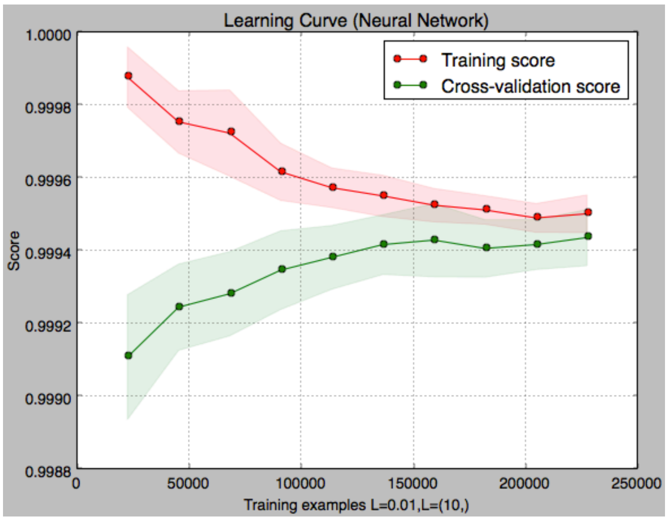


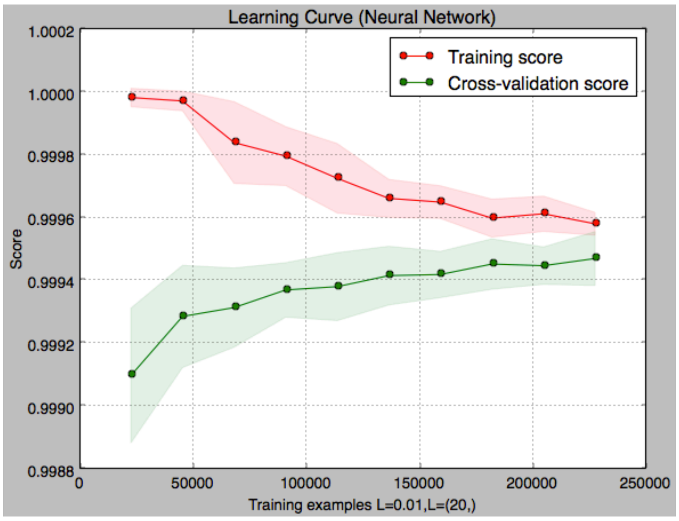


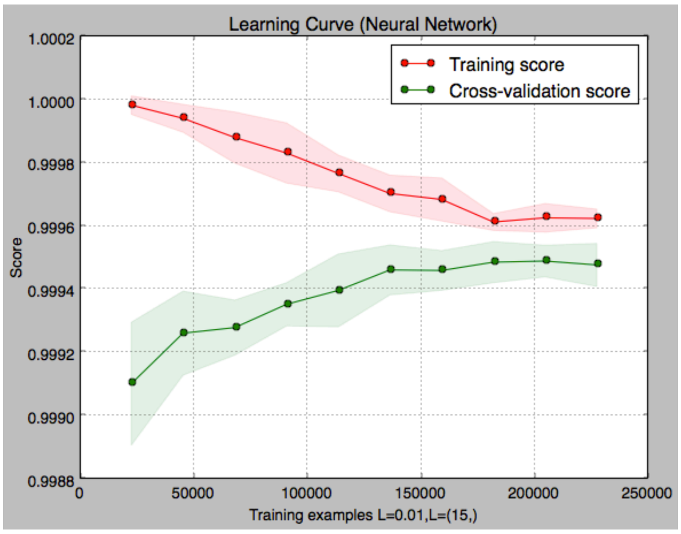


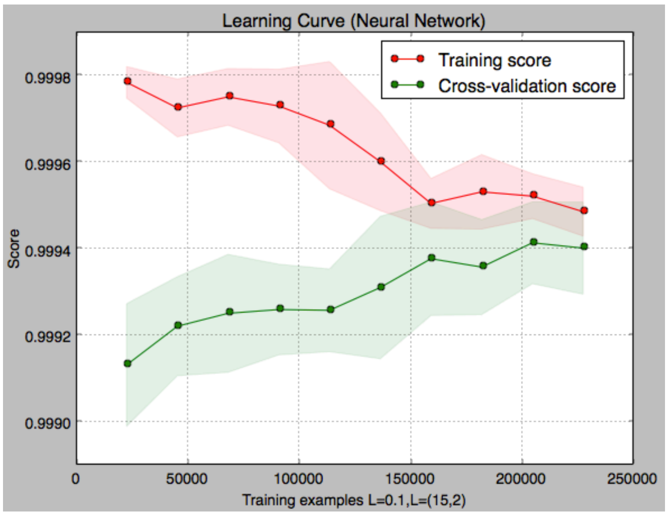


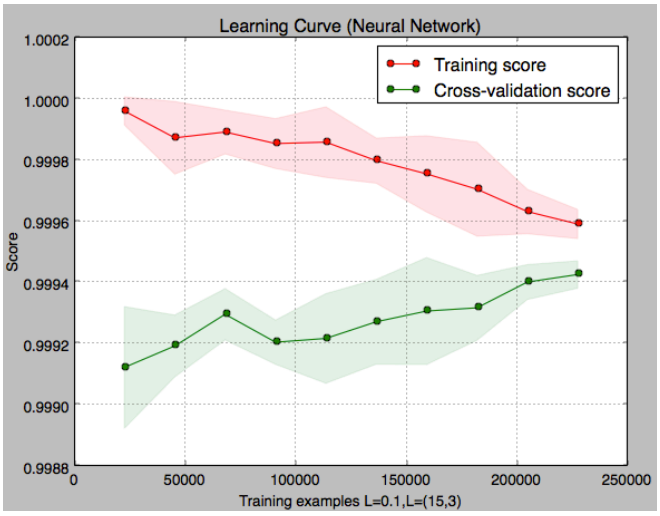
Card data











Several architectures are tested ((10,), (15,), (20,), (15,2), (15,3)). From the graphs above, we see that whether adding nodes to the single layer of adding more layers to estimate does not impact much on the final performance. There is really no “best” structure for solving our problems here.

Boosting (Decision Tree)

Cancer data:

(unpruned)

```
train_size | accu
   426     | 0.965
Classification report:
          precision   recall  f1-score   support

     0.0     0.97     0.98     0.97      91
     1.0     0.96     0.94     0.95      52

avg / total      0.96     0.97     0.96      143
```

(pruned)

```
train_size | accu
   426     | 0.972
Classification report:
          precision   recall  f1-score   support

     0.0     0.98     0.98     0.98      91
     1.0     0.96     0.96     0.96      52

avg / total      0.97     0.97     0.97      143
```

Card data:

(unpruned)

```
train_size | accu | node# | tree_dep
 213605  | 0.9984 |  303   |    21
Classification report:
          precision   recall  f1-score   support

     0.0     1.00     1.00     1.00    71081
     1.0     0.87     0.11     0.19     121

avg / total      1.00     1.00     1.00    71202
```

(pruned)

```
train_size | accu | node# | tree_dep
 213605  | 0.9984 |  113   |    10
Classification report:
          precision   recall  f1-score   support

     0.0     1.00     1.00     1.00    71081
     1.0     0.87     0.11     0.19     121

avg / total      1.00     1.00     1.00    71202
```

The pruning is done by setting the parameter of the DecisionTreeClassifier: min_impurity_decrease=0.01, which means that a node would be split only if this split decreases the impurity by at least 0.01.

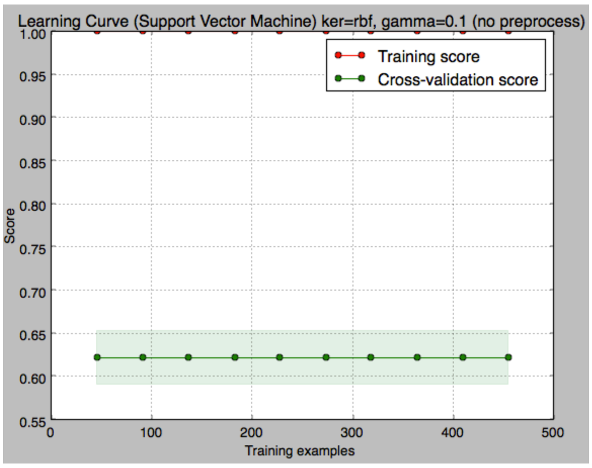
Similarly to the decision tree, the model accuracy for cancer data gets improved slightly while the impacts on the card data is small. The pruning can effectively avoid some overfitting caused by simulating the whole dataset.

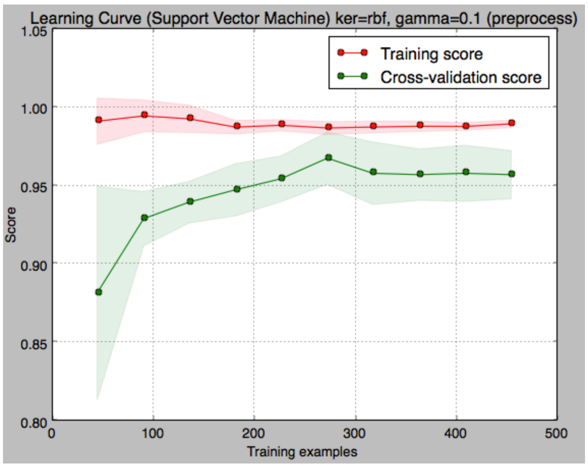
Support Vector Machine

(1) Data Preprocessing

(Cancer data used)

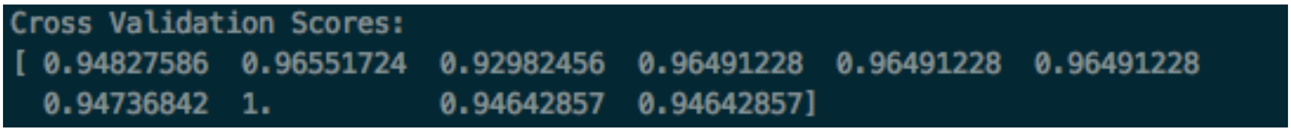
The scale of the data used matters for support vector machine, but it also depends on the kernel selected.





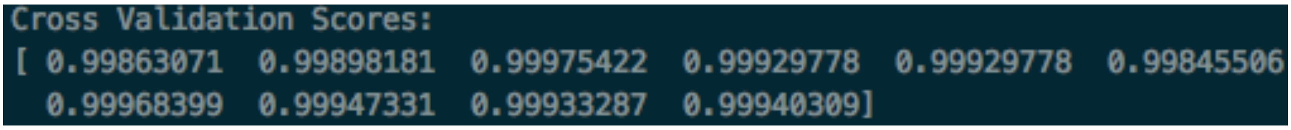
(2) Cross Validation

A 10 fold cross validation is performed for both datasets:
Cancer data:



```
Cross Validation Scores:
[ 0.94827586  0.96551724  0.92982456  0.96491228  0.96491228  0.96491228
  0.94736842  1.          0.94642857  0.94642857]
```

Card data:



```
Cross Validation Scores:
[ 0.99863071  0.99898181  0.99975422  0.99929778  0.99929778  0.99845506
  0.99968399  0.99947331  0.99933287  0.99940309]
```

Similarly, there are no preferences inside the datasets.

(3) Kernel and parameter analysis

3 kinds of kernels: a Radial Basis Function kernel with varying gamma, a polynomial kernel with different degrees and a linear kernel are tested.

Cancer data:

kernel	gamma/degree	accuracy	running_time	train_size	test_size
rbf	0.05	0.972027972028	0.008853912353	426	143
rbf	0.1	0.972027972028	0.008430004119	426	143
rbf	0.2	0.958041958042	0.022732019424	426	143
rbf	0.5	0.839160839161	0.021275997161	426	143
linear	/	0.944055944056	0.006678819656	426	143
poly	1	0.972027972028	0.006264925003	426	143
poly	2	0.783216783217	0.014128923416	426	143
poly	3	0.881118881119	0.014178037643	426	143

Card data:

kernel	gamma/degree	accuracy	running_time	train_size	test_size
rbf	0.1	0.998918569703	1398.22319603	213605	61202
rbf	0.2	0.998553411421	4600.30566788	213605	61202
rbf	0.5	0.998497233224	9203.99191213	213605	61202
linear	/	0.999410128929	85.847935915	213605	61202

(The computation of "poly" is beyond what my laptop could accomplish…)

In a "rbf" kernel, intuitively, gamma indicates how far the influence of 1 particular examples reaches. Low gamma values means "far" and high means "close". For the cancer data, as gamma increases, the accuracy drops. Based on the effects of gamma, a higher gamma value may limit the learning effectiveness of each sample, therefore the model with a higher gamma value could lose information of the important samples in the dataset, resulting in a lower accuracy.

The "poly" kernel with a greater degree generally turns to overfit the dataset, hence a lower accuracy as shown above. In practice, a best degree can be always tested out by a exhaustive method and observing the changes in accuracy. Note that a higher order "poly" kernel will increase the running time especially when the dataset is fairly large.

Frankly, the "LinearSVC" classifier provided by scikit-learn is a big surprise. It runs way more fast than other algorithm. For the card data, (1-85/1398)=94% time was saved while the performance was preserved. Whenever, in the future, a large dataset is considered, I will go for this kernel at first.

K-Nearest Neighbor

Cancer data:

K	accuracy	running_time	train_size	test_size
1	0.923076923077	0.00595998764038	426	143
2	0.923076923077	0.00464797019958	426	143
3	0.923076923077	0.0056631565094	426	143
5	0.93006993007	0.00699305534363	426	143
7	0.93006993007	0.00505185127258	426	143
10	0.93006993007	0.00529003143311	426	143
20	0.916083916084	0.00636696815491	426	143
30	0.909090909091	0.012579202652	426	143
50	0.902097902098	0.00762605667114	426	143

Card data:

K	accuracy	running_time	train_size	test_size
1	0.999297772534	91.6599030495	213605	61202
2	0.999227549788	107.564018011	213605	61202
3	0.999269683436	121.623170853	213605	61202
5	0.999255638887	142.292201042	213605	61202
7	0.999171371591	150.270209074	213605	61202
10	0.999101148844	163.601802111	213605	61202
20	0.999030926098	194.459421873	213605	61202
30	0.998960703351	226.733394861	213605	61202
50	0.998904525154	325.014518976	213605	61202

First, the running time increases as k gets larger. This is not so obvious for the cancer data because the dataset size is not so large and in the testing range of k, all kernels runs fairly fast.

For the cancer data, as k increases, the accuracy improves a little from 1 to 10, then drops gradually. In this case, k=10 may indicate the turning point where the model starts to overfit the dataset. When k gets really big, some nonsensical "nearest neighbors" get involved in making decisions, adding errors. In a practical occasion, the best k value can be determined by a similar table.

Attribution

This whole project is built using python 2.7 with

- Python machine learning library *scikit-learn* (version 0.18)
- Scientific computing package *numpy*
- 2D plotting library *matplotlib*