STAT 8330 FALL 2015 ASSIGNMENT 6

Peng Shao

October 25, 2015

(Most codes and plots are listed in the appendices)

▶ **Exercises 1.** **Solution.**

I fit most of models except DBN in this homework by using `train()` function in `caret` package, because it can use multiple cores to execute high performance computation, and it can easily realize cross valitadion and model comparison.

For this problem, I refit the `Carseats` dataset using gradient boosting tree, bagging tree, random forest and single-layer neural network with 10 fold cross validation. The performances of models based on training datasets are

```
summary(resamps.Carseats)
```

```
##         Length Class      Mode
## call    2      -none-     call
## values  9      data.frame list
## models  4      -none-     character
## metrics 2      -none-     character
## timings 3      data.frame list
## methods 4      -none-     character
```

There is no doubt that the neural network should be the best model. The corresponding test MSEs are

```
mse.Carseats
```

```
##              ANN     randomForest GradientBoosting       BaggedCART
##         1.118744         2.685115         1.534656         3.453284
```

For neural network, I search the tuning parameters from

```
nnet.grid <- expand.grid(.decay = c(0.5, 0.1, 0.005, 0.001),
                         .size = 1:10)
```

and the best hyper parameters of the neural network are

```
nnet.fit.Carseats$bestTune
```

```
##   size decay
## 1    1 0.001
```

Then I try the `nnet()` function in `nnet` package to refit the net, using the tuning parameters above. The test MSE of the refitting neural is

```
library(nnet)
nnet.fit.Carseats.2 <- nnet(Sales ~ . , data = Carseats.train,
                            size=1, decay=0.001, linout = TRUE, trace = FALSE)
nnet.pred.Carseats.2 <- predict(nnet.fit.Carseats.2, Carseats.test, type="raw")
rmse <- mean((nnet.pred.Carseats.2 - Carseats.test$Sales)^2)
rmse
```

```
## [1] 6.366908
```

This is a little different from previouse model, which may indicate overfitting problems, since the dataset is not so “large”.

▶ **Exercises 2. Solution.**

The method for this problem is identical to the previous problem. The performances of the four models based on training datasets are

```
summary(resamps.pima)
```

```
##
## Call:
## summary.resamples(object = resamps.pima)
##
## Models: ANN, randomForest, GradientBoosting, BaggedCART
## Number of resamples: 10
##
## Accuracy
##                  Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## ANN              0.60  0.7500 0.7560 0.7601  0.7974 0.8571    0
## randomForest     0.65  0.7000 0.7321 0.7198  0.7500 0.7500    0
## GradientBoosting 0.65  0.7431 0.7947 0.7745  0.8375 0.8571    0
## BaggedCART       0.65  0.6625 0.7256 0.7251  0.7875 0.8000    0
##
## Kappa
##                    Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## ANN              0.1209  0.3566 0.4483 0.4505  0.5657 0.6667    0
## randomForest     0.1094  0.2975 0.3414 0.3392  0.4176 0.4681    0
## GradientBoosting 0.2045  0.4090 0.4869 0.4745  0.6080 0.6897    0
## BaggedCART       0.2045  0.2060 0.3351 0.3613  0.5141 0.5604    0
```

The random forest is the best model while the neural network becomes the worst one. But for the corresponding test error rates are

```
er.pima
```

```
##              ANN     randomForest GradientBoosting       BaggedCART
##        0.1987952        0.2379518        0.2138554        0.2409639
```

So the neural network has the lowest test error rate. Just to be cautious, the lowest error rate here has nothing to do with stability. It is very likely this neural network will perform poorly on other test data sets

The best hyper parameters of the neural network are

```
nnet.fit.pima$bestTune
```

```
##    size decay
## 32    2   0.5
```

Then I try the `nnet()` function in `nnet` package to refit the net, using the tuning parameters above. The test error of the refitting neural is

```
library(nnet)
nnet.fit.pima.2 <- nnet(type ~ . , data = Pima.tr,
                        size=2, decay=0.5, trace = FALSE)
nnet.pred.pima.2 <- predict(nnet.fit.pima.2, Pima.tr)
rmse <- mean((nnet.pred.Carseats.2 - Carseats.test$Sales)^2)
rmse
```

```
## [1] 6.366908
```

This is a little different from previouse model, which may indicate overfitting problems, since the dataset is not so “large”.

- **Exercises 3. Solution.**
- **Exercises 4. Solution.**
- **Exercises 5. Solution.**

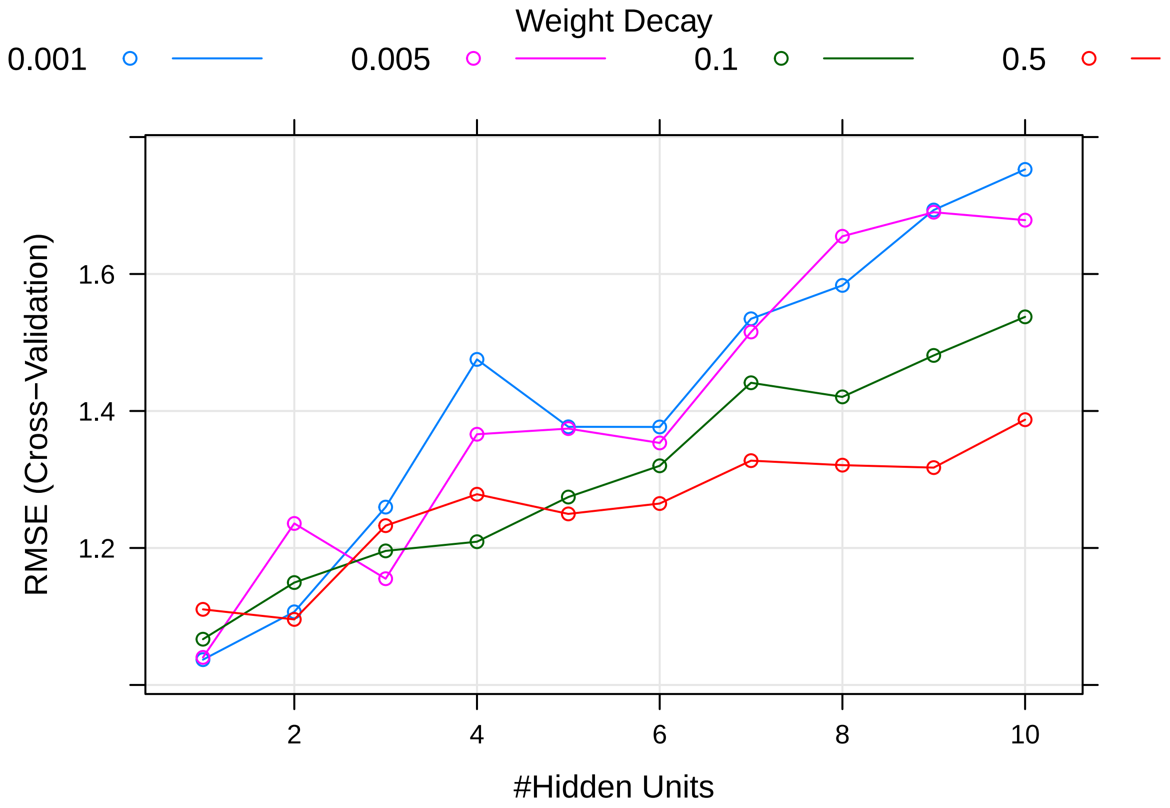
APPENDICES

Code

```
time13-time1
```

```
## Time difference of 9.986183 mins
```

Plot


Weight Decay
RMSE (Cross−Validation)
#Hidden Units