Homework 6

Aaron Politsky

November 8, 2015

```
set.seed(99)
library(h2o)
```

```
## Loading required package: statmod
##
## ----------------------------------------------------------------------
##
## Your next step is to start H2O:
##     > h2o.init()
##
## For H2O package documentation, ask for help:
##     > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit http://docs.h2o.ai
##
## ----------------------------------------------------------------------
##
##
## Attaching package: 'h2o'
##
## The following objects are masked from 'package:stats':
##
##     sd, var
##
## The following objects are masked from 'package:base':
##
##     %*%, %in%, apply, as.factor, as.numeric, colnames, colnames<-,
##     ifelse, is.factor, is.numeric, log, trunc
```

```
load("data.Rda")
source("~/HelpR/lift.R")
source("~/HelpR/EvaluationMetrics.R")

#source("ParseData.R")
#data <- parse_human_activity_recog_data()
```

1. Build a Neural Network

```
# start or connect to h2o server
h2oServer <- h2o.init(max_mem_size="4g", nthreads=-1)
```

```
## Successfully connected to http://127.0.0.1:54321/
```

```
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         7 hours 28 minutes
##     H2O cluster version:        3.5.0.3232
##     H2O cluster name:           H2O_started_from_R_aaron_pvl873
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   3.56 GB
##     H2O cluster total cores:    4
##     H2O cluster allowed cores:  4
##     H2O cluster healthy:        TRUE
```

```r
# we need to load data into h2o format
train_hex = as.h2o(data.frame(x=data$X_train, y=data$y_train))
```

```
##
  |
  |                                                                 |   0%
  |
  |=================================================================| 100%
```

```r
test_hex = as.h2o(data.frame(x=data$X_test, y=data$y_test))
```

```
##
  |
  |                                                                 |   0%
  |
  |=================================================================| 100%
```

```r
predictors <- 1:(ncol(train_hex)-1)
response <- ncol(train_hex)
```

Let's see how different models perform when we try different parameters.

```r
hyper.params <-
  list(
    epochs=c(2,5,10),
    hidden=list(c(64), c(128), c(256), c(512), c(1024),
                c(256,256), c(1024,1024), c(128,128,128))
  )
```

```r
dl.grid <- h2o.grid(
  algorithm = "deeplearning",
  x=predictors, y=response,
  training_frame=train_hex,
  activation="Tanh",
  classification_stop=-1,  # Turn off early stopping
  l1=1e-5,
  hyper_params = hyper.params
)
summary(dl.grid)
dl.grid.models <- lapply(dl.grid@model_ids, function(id) h2o.getModel(id))
model.paths <- lapply(dl.grid.models, function(m) h2o.saveModel(m, path="models"))
```

```
# performance on test set
ptest.list <- lapply(dl.grid.models, function(m) h2o.performance(m, test_hex))
cm.test.list <- lapply(ptest.list, function(ptest) h2o.confusionMatrix(ptest))
```

Which did the best?

```
library(plyr)
ptest.df <- ldply(cm.test.list,
                  function(cm)
                    c(total.error.rate = cm$Error[7]))
best.model.index <- which.min(ptest.df$total.error.rate)
best.dl.model <- dl.grid.models[[best.model.index]]
cm.test.list[[best.model.index]]
```

```
## Confusion Matrix - (vertical: actual; across: predicted): vertical: actual; across: predicted
##                    Laying Sitting Standing Walking WalkingDownstairs
## Laying                513       0       24       0                 0
## Sitting                 0     425       63       0                 0
## Standing                0      10      521       1                 0
## Walking                 0       0        0     491                 4
## WalkingDownstairs       0       0        2       4               403
## WalkingUpstairs         0       0        1      25                16
## Totals                513     435      611     521               423
##                    WalkingUpstairs  Error           Rate
## Laying                           0 0.0447 =    24 / 537
## Sitting                          3 0.1344 =    66 / 491
## Standing                         0 0.0207 =    11 / 532
## Walking                          1 0.0101 =     5 / 496
## WalkingDownstairs               11 0.0405 =    17 / 420
## WalkingUpstairs                429 0.0892 =    42 / 471
## Totals                         444 0.0560 = 165 / 2,947
```

Model 2 did the best, with a **5.5989%** test error rate.