北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R
语言数据分析

既是世间法、自当有分别

艾新波 / 2018 • 北京

课程体系

- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
  -
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

R语言实现

nnet(nnet) R Documentation

Fit Neural Networks

Description

Fit single-hidden-layer neural network, possibly with skip-layer connections

Usage

nnet(x,…)

… arguments passed to or form other methods.

Details

If the response in formula is a factor,an appropriate classification network is constructed;this has one output and entropy fit if the number of levels is two,and a number of outputs equal to the number classes and a softmax output stage for more levels.If the response is not a factor,it is passed on unchanged to nnet.default.

Optimization is done via the BFGS method of optim.

R语言实现

nnet::nnet部分参数设置:

Arguments

- **size:** number of units in the hidden layer.
- **Wts:** initial parameter vector. If missing chosen at random.
- **rang:** Initial random weights on [-rang, rang]. Value about 0.5 … so that rang * max(|x|) is about 1.
- **decay:** parameter for weight decay. Default 0.
- **maxit:** maximum number of iterations. Default 100.
- **MaxNWts:** The maximum allowable number of weights.

R语言实现

```
library(nnet)
set.seed(2012)
imodel <- nnet(wlfk~., data = cjb[train_set_idx, ], size = 7)
names(imodel)
#> [1] "n"             "nunits"         "nconn"
#> [4] "conn"          "nsunits"        "decay"
#> [7] "entropy"       "softmax"        "censored"
#> [10] "value"         "wts"            "convergence"
#> [13] "fitted.values" "residuals"      "lev"
#> [16] "call"          "terms"          "coefnames"
#> [19] "contrasts"     "xlevels"
```

R语言实现

```
imodel$n
#> [1] 10  7  1
imodel$wts
#> [1]   -0.394367962    0.341672486   -0.305656476
……
#> [85]    0.019783268
imodel$fitted.values
#      [,1]
# 1   0.8048857
……
# 541 0.2047307
```

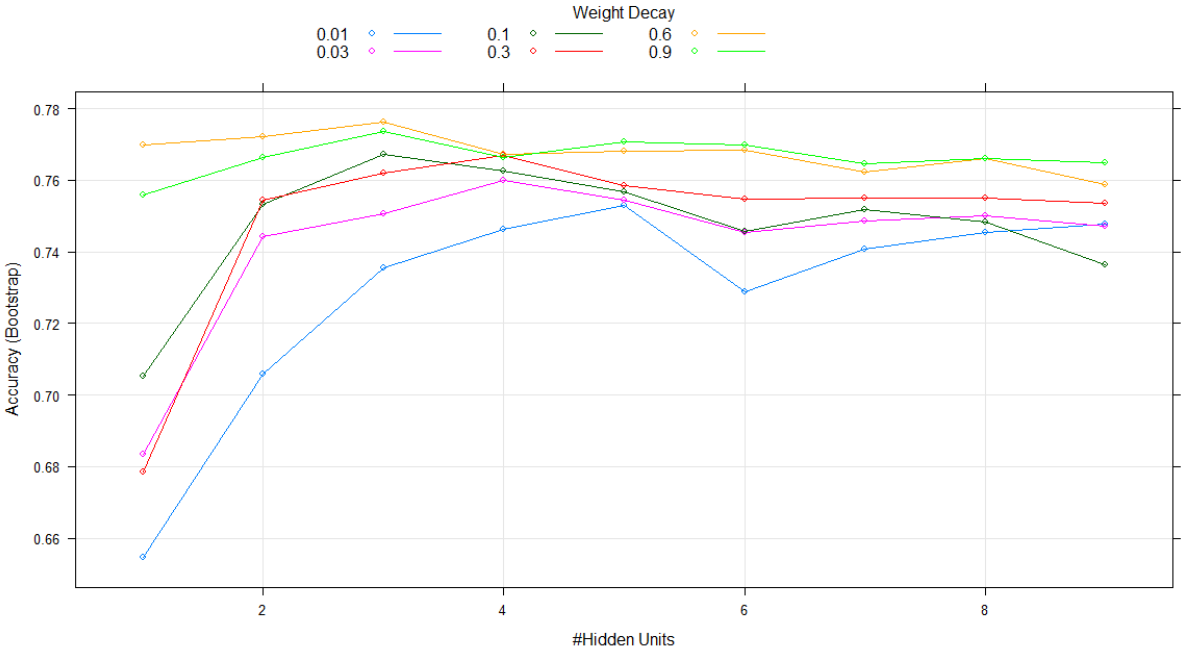
R语言实现

```
predicted_train <- predict(imodel,type = "class",
                           newdata = cjb[train_set_idx,])
Metrics::ce(cjb$wlfk[train_set_idx], predicted_train)
#> [1] 0.1996303
predicted_test <- predict(imodel, type = "class",
                          newdata = cjb[-train_set_idx,])
Metrics::ce(cjb$wlfk[-train_set_idx], predicted_test)
#> [1] 0.1759657
```

R语言实现

```
library(caret)#通过caret包来进行格子点搜索
set.seed(2012)
nn_grid <- expand.grid(
  size = 1:9, decay = c(0.03, 0.1, 0.3, 0.6, 0.9))
imodel <- train(wlfk ~ ., data = cjb,
  method = "nnet", maxit = 2000,
  tuneGrid = nn_grid)
imodel$bestTune
#>      size decay
#> 17    3   0.6
plot(imodel)#查看训练结果
```

R语言实现


Weight Decay
0.01
0.03
0.1
0.3
0.6
0.9
Accuracy (Bootstrap)
0.78
0.76
0.74
0.72
0.70
0.68
0.66
2
4
6
8
#Hidden Units

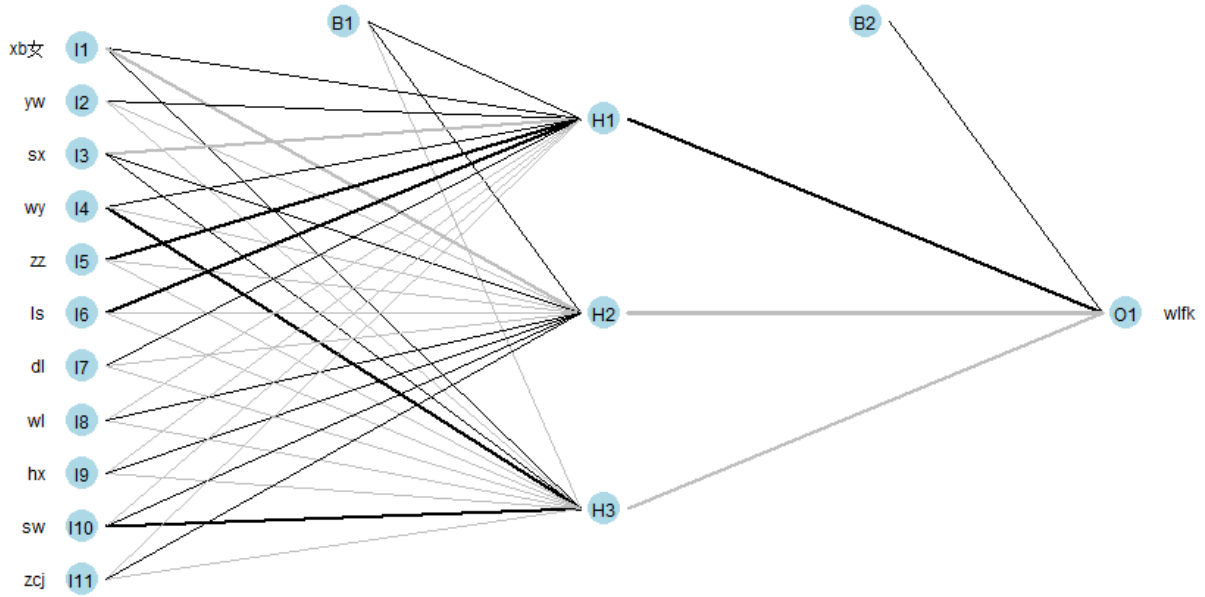
R语言实现

```
predicted_train <- predict(imodel,
    newdata = cjb[train_set_idx,], type = "raw")
Metrics::ce(cjb$wlfk[train_set_idx],
            predicted_train)
#> [1] 0.1848429
predicted_test <- predict(imodel,
    newdata = cjb[-train_set_idx,], type = "raw")
Metrics::ce(cjb$wlfk[-train_set_idx],
            predicted_test)
#> [1] 0.1545064
```

R语言实现

```
#绘制神经网络
library(NeuralNetTools)
imodel2 <-  nnet(
  wlfk ~ ., data = train_set,
  decay = imodel$bestTune$decay, size = imodel$bestTune$size,
  maxit = 2000)
plotnet(
  imodel2, rel_rsc = c(1.8,3),
  circle_cex = 3, cex_val = 0.75,
  bord_col = "lightblue", max_sp = TRUE)
```

R语言实现


B1
B2
xb女
I1
yw
I2
sx
I3
wy
I4
zz
I5
ls
I6
dl
I7
wl
I8
hx
I9
sw
I10
zcj
I11
H1
H2
H3
O1
wlfk

R语言实现

```
sp <- Sys.time() #记录开始时间
cat("\n[Start at:", as.character(sp))
for (i in 1:length(kfolds)) {
  curr_fold <- kfolds[[i]] #当前这一折
  train_set <- cjb[-curr_fold,] #训练集
  test_set <- cjb[curr_fold,] #测试集
  imodel_kfold <-  nnet(
    wlfk ~ ., data = train_set, maxit = 2000,
    decay = imodel$bestTune$decay,
    size = imodel$bestTune$size)
  predicted_train <-  predict(imodel_kfold,
                              train_set, type = "class")
```

R语言实现

```
  imetrics("nnet", "Train",
           predicted_train, train_set$wlfk)
  predicted_test <- predict(imodel_kfold,
                            test_set, type = "class")
  imetrics("nnet", "Test",
           predicted_test, test_set$wlfk)
}
ep <- Sys.time()
cat("\tFinised at:", as.character(ep), "]\n")
cat("[Time Ellapsed:\t",
    difftime(ep, sp, units = "secs"), " seconds]\n")
```

R语言实现

```
#> 101          nnet Train 0.8175287  0.1824713
#> 102          nnet  Test 0.8461538  0.1538462
#> 109          nnet Train 0.8048780  0.1951220
#> 110          nnet  Test 0.8181818  0.1818182
#> 111          nnet Train 0.7962697  0.2037303
#> 112          nnet  Test 0.8181818  0.1818182
.....
#> 117          nnet Train 0.8206600  0.1793400
#> 118          nnet  Test 0.7792208  0.2207792
#> 119          nnet Train 0.8077475  0.1922525
#> 120          nnet  Test 0.7662338  0.233766
```

谢谢聆听

Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ： 23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics