

APA - Entrega 5

Josep de Cid

Gener 2018

Problema 8. Permeabilitat de roquetes amb la xarxa MLP [R]
Es disposa de 48 mesures de roques d'un dipòsit de petroli. L'objectiu és modelar la permeabilitat en funció de l'àrea, el perímetre i la forma. En primer lloc transformem les dades per ajudar a l'ajust del model:

```
library(datasets)
data(rock)
?rock

rock.x <- data.frame(area = scale(rock$area),
                     perim = scale(rock$peri),
                     shape = scale(rock$shape))
rock.y <- log(rock$perm)
```

Entreneu una xarxa MLP per aprendre la tasca. Donat el baix número d'exemples, useu *leave-one-out cross-validation* i regularització per trobar la millor xarxa. Per avaluar el model, feu una gràfica de resposta predita vs. observada i guieu-vos per l'error quadràtic predictiu.

En primer lloc, farem la transformació, però amb unes petites variacions respecte la transformació de l'enunciat, per preferències personals en l'estil de programació, de forma que unim totes les dades ja transformades en el mateix *dataframe*.

Un cop tenim les dades transformades, passem a entrenar la xarxa neuronal, definint primer l'objecte *trainControl*, amb el mètode **LOOCV** (Leave-One-Out Cross-Validation). Entrenarem el model amb el camp *perm* depenent de la resta, òbviament posant com a mètode *nnet* i indicant que és un problema de regressió. Com a mètrica farem servir **RMSE** (Root Mean Square Error) i incrementarem el nombre d'iteracions màximes, que per defecte és 100, a 2000. Per a provar diferent nombre de neurones i *decay*, definim un grid amb neurones entre 1 i 15, i *decay* entre 10^{-3} i 1.

Observant la variable `nnet.model$bestTune` i el RMSE, podem veure que en l'arquitectura òptima obtinguda hi ha només dues neurones a la capa oculta amb `decay = 0.03981072`, obtenint un $RMSE = 0.8133283$.

```

library(datasets)
library(caTools)
library(ggplot2)
library(caret)
library(doMC)

registerDoMC(cores = 4)

data(rock)
rock.df <- data.frame(area = scale(rock$area),
                      perm = scale(rock$peri),
                      shape = scale(rock$shape),
                      perm = log(rock$perm))

split = sample.split(rock.df$perm, SplitRatio = 0.8)
training.set = subset(rock.df, split == TRUE)
test.set = subset(rock.df, split == FALSE)

trc <- trainControl(method = 'LOOCV')
sizes <- seq(1, 15, by = 1)
decays <- 10^seq(-3, 0, by = 0.2)
nnet.model <- train(perm ~ .,
                   data = training.set,
                   linout = TRUE, trace = FALSE,
                   method = 'nnet', metric = 'RMSE',
                   trControl = trc, maxit = 2000,
                   tuneGrid = expand.grid(.size = sizes,
                                           .decay = decays))

pred <- predict(nnet.model, newdata = test.set)

(best <- nnet.model$bestTune)
(rmse <- nnet.model$results[as.numeric(row.names(best)),]$RMSE)

ggplot(test.set, aes(x = perm, y = pred)) +
  geom_point() +
  geom_smooth(method = 'lm', col = 'darkred') +
  ggtitle('Actual vs Predicted (NN)') +
  theme(legend.position="none") +
  xlab('Actual') + ylab('Predicted')

```

Actual vs Predicted (NN)

