Universidade Federal de Minas Gerais
Aluno: Giovanni Martins de Sá Júnior
Matrícula: 2017001850

Exercício 8: Redes Neurais Artificiais

Nesta oitava atividade, buscou-se determinar a viabilidade no uso de modelos RBF, com ou sem clustering para a determinação dos centros para a resolução de problemas reais. O primeiro problema que nos propusemos a resolver, é o da base Breast Cancer (diagnostic), enquanto o segundo é o da base Statlog (Heart). Após feito isso, os resultados são comparados com os obtidos para ELMs, com as mesmas bases.

Breast Cancer

Primeiramente, foi utilizado o código apresentado no final desta seção para realizar diferentes treinamentos de redes RBF com tamanhos variados e técnicas variadas de modo a comparar a sua performance. Na Tabela 1, é possível ver a acurácia tanto no conjunto de treinamento quanto no de testes para diferentes números de centros p.

Para isso, foram utilizadas duas técnicas: na primeira, os centros foram escolhidos aleatoriamente, ao escolher aleatoriamente dois dos pontos de entrada, sendo que cada centro seria a média entre eles e o raio correspondente a esse centro a distância entre eles. Na segunda, utilizou-se o algoritmo k-médias para determinar os centros e matrizes ótimos de acordo com a clusterização do conjunto de entrada.

p	rnd_train_acc	rnd_test_acc	km_train_acc	km_test_acc
5	0.833 ± 0.118	0.843 ± 0.141	0.822 ± 0.031	0.835 ± 0.059
10	0.874 ± 0.079	0.899 ± 0.092	0.751 ± 0.068	0.704 ± 0.085
30	0.861 ± 0.106	0.869 ± 0.122	0.697 ± 0.066	0.663 ± 0.074
50	0.862 ± 0.105	0.876 ± 0.126	0.685 ± 0.016	0.638 ± 0

Tabela 1 – Acurácias das Redes RBF para diferentes números de centros

Como foi possível observar, a rede não tem boa performance com o k-means, sugerindo que a clusterização espacial não é representativa das classes estudadas. Em realidade, a pior performance sugere que os clusters enganam quanto à sua natureza.

Além disso, o fato de que os dados não se agrupam em clusters faz com que a rede RBF não seja necessariamente a melhor forma de solucionar esse problema. De fato, com 50 neurônios de mapeamento intermediário uma rede RBF consegue 96% de acurácia nesse mesmo problema, o que sugere que a separação do espaço não se dá por meio de funções gaussianas/radiais, mas com algum outro tipo de superfície de separação melhor compreendido pela ELM.

A seguir é mostrada a implementação realizada:

```
<<>>=
data("BreastCancer")
prepareBCInput <- function(data) {

  data <- data %>% mutate_all(funs(ifelse(is.na(.), 0, .)))

  input_data <- cbind(
    data$Cl.thickness,
    data$Cell.size,
    data$Cell.shape,
    data$Marg.adhesion,
    data$Epith.c.size,
    data$Bare.nuclei,
    data$Bl.cromatin,
    data$Normal.nucleoli,
    data$Mitoses
  )
  return (input_data)
}

# Separacao dos Dados
TRAIN_PERCENTAGE <- 0.7
SAMPLE_SIZE <- 699

BENIGN <- 0
MALIGNANT <- 1

train_rows <- sample(SAMPLE_SIZE, size=floor(TRAIN_PERCENTAGE*SAMPLE_SIZE))
train_data <- BreastCancer[train_rows,]
test_data <- BreastCancer[-train_rows,]

# Aplicacao do Modelo

x_train <- prepareBCInput(train_data)

y_train <- as.matrix(
  as.numeric(
    mapvalues(train_data[,11], from=c('benign', 'malignant'), to=c(BENIGN, MALIGNANT))
  ) - replicate(floor(TRAIN_PERCENTAGE*SAMPLE_SIZE), 1),
ncol=1)

x_test <- prepareBCInput(test_data)
y_test <- as.matrix(
  as.numeric(
    mapvalues(test_data[,11], from=c('benign', 'malignant'), to=c(BENIGN, MALIGNANT))
  ) - replicate(dim(test_data)[[2]], 1),
ncol=1)
```

```r
numbers_centers <- c(5, 10, 30, 50)
number_trainings <- 10

acc_list <- list()

activ_func <- function(y) { return (1*(y>=0.5)) }

for (i in 1:length(numbers_centers)) {
  p <- numbers_centers[i]
  rnd_train_acc_list <- list()
  rnd_test_acc_list <- list()
  km_train_acc_list <- list()
  km_test_acc_list <- list()
  for (j in 1:number_trainings) {
    # Random
    random_rbf_centers <- random_centers(p, x_train)
    random_rbf_params <- train_rbf(x_train, y_train, p, FALSE,
                                   random_rbf_centers[[1]],
                                   random_rbf_centers[[2]])

    # Train
    y_hat_random_train <- YRBF(x_train, random_rbf_params[[1]],
                               random_rbf_params[[2]], random_rbf_params[[3]],
                               activ_func)
    rnd_train_results <- confusionMatrix(data=factor(y_hat_random_train),
                                         reference=factor(y_train))
    rnd_train_acc_list[j] <- rnd_train_results$overall[1]
    # Test
    y_hat_random_test <- YRBF(x_test, random_rbf_params[[1]],
                               random_rbf_params[[2]], random_rbf_params[[3]],
                               activ_func)
    rnd_test_results <- confusionMatrix(data=factor(y_hat_random_test),
                                         reference=factor(y_test))
    rnd_test_acc_list[j] <- rnd_test_results$overall[1]

    # Kmeans
    km_rbf_params <- train_rbf(x_train, y_train, p, TRUE)

    # Train
    y_hat_km_train <- YRBF(x_train, km_rbf_params[[1]],
                               km_rbf_params[[2]], km_rbf_params[[3]],
                               activ_func)
    km_train_results <- confusionMatrix(data=factor(y_hat_km_train),
                                         reference=factor(y_train))
    km_train_acc_list[j] <- km_train_results$overall[1]
```

```
  # Test
  y_hat_km_test <- YRBF(x_test, km_rbf_params[[1]],
                            km_rbf_params[[2]], km_rbf_params[[3]],
                            activ_func)
  km_test_results <- confusionMatrix(data=factor(y_hat_km_test),
                                     reference=factor(y_test))
  km_test_acc_list[j] <- km_test_results$overall[1]
}
rnd_train_acc <- paste(
  round(mean(as.numeric(rnd_train_acc_list)),3),
  "\U00B1",
  round(sqrt(var(as.numeric(rnd_train_acc_list))), 3)
)
rnd_test_acc <- paste(
  round(mean(as.numeric(rnd_test_acc_list)),3),
  "\U00B1",
  round(sqrt(var(as.numeric(rnd_test_acc_list))), 3)
)
km_train_acc <- paste(
  round(mean(as.numeric(km_train_acc_list)),3),
  "\U00B1",
  round(sqrt(var(as.numeric(km_train_acc_list))), 3)
)
km_test_acc <- paste(
  round(mean(as.numeric(km_test_acc_list)),3),
  "\U00B1",
  round(sqrt(var(as.numeric(km_test_acc_list))), 3)
)

acc_list[[i]] <- list(p, rnd_train_acc,rnd_test_acc, km_train_acc, km_test_acc)
```

Heart

Nesta seção, realizamos uma formulação análoga à anterior. Nesse caso, o código utilizado é apresentado no final da sessão. Novamente, avaliamos a performance da rede para diferentes dimensionamentos e coletamos as acurácias com e sem o uso de k-médias. Os resultados podem ser vistos na Tabela 2. A forma de escolha dos centros aleatórios foi a mesma adotada anteriormente.

p	rnd_train_acc	rnd_test_acc	km_train_acc	km_test_acc
5	0.529 ± 0	0.617 ± 0	0.529 ± 0	0.617 ± 0
10	0.529 ± 0	0.617 ± 0	0.537 ± 0.008	0.617 ± 0
30	0.529 ± 0	0.617 ± 0	0.55 ± 0.026	0.617 ± 0
50	0.529 ± 0	0.617 ± 0	0.602 ± 0.026	0.617 ± 0
100	0.531 ± 0.005	0.617 ± 0	0.819 ± 0.016	0.617 ± 0

Tabela 2 – Acurácias das Redes RBF para diferentes números de centros

Como pode ser visto na tabela acima, a rede não tem de modo geral, uma boa performance, sugerindo novamente uma deslocalização espacial da resposta. A acurácia média de treinamento mais alta para o caso médias com 100 centros pode ser simplesmente uma

anomalia, considerando que o número de centros passa a se aproximar do número de amostras, o que torna o modelo excessivamente específico.

Ao comparar esses resultados pelos produzidos pela rede ELM, tem-se que a performance das ELMs continua melhor, mas nunca passa de 80%, o que sugere que nenhuma das duas formas de rede é a mais adequada para solucionar esse problema.

A seguir é mostrada a implementação realizada:

```
<<>>=
heart <- as.matrix(read.table('heart.dat'))
heart[is.na(heart)] <- 0

# DATA SEPARATION
TRAIN_PERCENTAGE <- 0.7
SAMPLE_SIZE <- 270

BENIGN <- 0
MALIGNANT <- 1

train_rows <- sample(SAMPLE_SIZE, size=floor(TRAIN_PERCENTAGE*SAMPLE_SIZE))
train_data <- heart[train_rows,]
test_data <- heart[-train_rows,]

# MODEL APPLICATION
x_train <- train_data[,1:13]
y_train <- train_data[,14] - 1

x_test <- test_data[,1:13]
y_test <- test_data[,14] - 1

numbers_centers <- c(5, 10, 30, 50, 100)
number_trainings <- 10

acc_list_2 <- list()
activ_func <- function(y) { return (1*(y>=0.5)) }
```

```
for (i in 1:length(numbers_centers)) {
  p <- numbers_centers[i]
  rnd_train_acc_list <- list()
  rnd_test_acc_list <- list()
  km_train_acc_list <- list()
  km_test_acc_list <- list()
  for (j in 1:number_trainings) {
    # Random
    random_rbf_centers <- random_centers(p, x_train)
    random_rbf_params <- train_rbf(x_train, y_train, p, FALSE,
                                   random_rbf_centers[[1]],
                                   random_rbf_centers[[2]])

    # Train
    y_hat_random_train <- YRBF(x_train, random_rbf_params[[1]],
                               random_rbf_params[[2]], random_rbf_params[[3]],
                               activ_func)
    rnd_train_results <- confusionMatrix(data=factor(y_hat_random_train),
                                         reference=factor(y_train))
    rnd_train_acc_list[j] <- rnd_train_results$overall[1]
    # Test
    y_hat_random_test <- YRBF(x_test, random_rbf_params[[1]],
                              random_rbf_params[[2]], random_rbf_params[[3]],
                              activ_func)
    rnd_test_results <- confusionMatrix(data=factor(y_hat_random_test),
                                        reference=factor(y_test))
    rnd_test_acc_list[j] <- rnd_test_results$overall[1]
```

```
    # Test
    y_hat_km_test <- YRBF(x_test, km_rbf_params[[1]],
                              km_rbf_params[[2]], km_rbf_params[[3]],
                              activ_func)
    km_test_results <- confusionMatrix(data=factor(y_hat_km_test),
                                       reference=factor(y_test))
    km_test_acc_list[j] <- km_test_results$overall[1]
  }
  rnd_train_acc <- paste(
    round(mean(as.numeric(rnd_train_acc_list)),3),
    "\U00B1",
    round(sqrt(var(as.numeric(rnd_train_acc_list))), 3)
  )
  rnd_test_acc <- paste(
    round(mean(as.numeric(rnd_test_acc_list)),3),
    "\U00B1",
    round(sqrt(var(as.numeric(rnd_test_acc_list))), 3)
  )
  km_train_acc <- paste(
    round(mean(as.numeric(km_train_acc_list)),3),
    "\U00B1",
    round(sqrt(var(as.numeric(km_train_acc_list))), 3)
  )
  km_test_acc <- paste(
    round(mean(as.numeric(km_test_acc_list)),3),
    "\U00B1",
    round(sqrt(var(as.numeric(km_test_acc_list))), 3)
  )

  acc_list_2[[i]] <- list(p, rnd_train_acc,rnd_test_acc, km_train_acc, km_test_acc)
}
```