UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Exercícios Computacionais 4: Radial Basis Function Networks

Aluno: *Pedro Vinícius A. B. de Venâncio*

Disciplina: *Redes Neurais Artificiais (EEE950)* – Professor: *Antônio de Pádua Braga*
Data: *Setembro de 2019*

Introdução

Este relatório tem como objetivo desenvolver[1] a rede neural artificial do tipo *Radial Basis Function (RBF)*. Tal modelo introduziu uma camada escondida entre as camadas de entrada e saída habituais de arquiteturas como *perceptron* e *adaline*, ampliando as possibilidades e contribuindo bastante com a área de aprendizado de máquina.

Radial Basis Function networks (RBFs)

As *Radial Basis Function networks*, desenvolvidas por David S. Broomhead e David Lowe (1988), contêm em sua arquitetura três camadas de neurônios: camada de entrada, camada escondida e camada de saída. Em particular, a camada escondida, cujos neurônios utilizam funções de base radiais, agrupa os dados de entrada em *clusters*, com a finalidade de tornar as entradas linearmente separáveis e, consequentemente, realizar o mapeamento $\boldsymbol{A_1}$ entre a camada de entrada e a camada escondida.

```
# Mapping from input layer to hidden layer
A_1 <- matrix(nrow = n_samples, ncol = k_clusters)
for(i in 1:n_samples){
  for(j in 1:k_clusters){
    v <- norm(X[i,] - as.matrix(centroid[j,]))
    A_1[i,j] <- gaussian(v, sigma)
  }
}
A_1 <- cbind(1, A_1)
```

A função de base radial mais utilizada nas redes RBF é a função Gaussiana (1):

$$\psi(v) = e^{-v/2\sigma^2} \tag{1}$$

onde σ é a largura da função radial e $v = \|\boldsymbol{X} - \boldsymbol{\mu}\|$ geralmente é dado pela distância Euclidiana da entrada $\boldsymbol{X}$ para o centro da função radial $\boldsymbol{\mu}$.

```
# Gaussian activation function
  gaussian <- function(v, sigma){
    return(exp(-v/(2*sigma^2)))
  }
```

[1] Todas as implementações foram feitas em linguagem R.

Os parâmetros da função Gaussiana são determinados por diversas metodologias. Entretanto, algumas são mais convenientes e têm sido efetivas na prática. O primeiro parâmetro a ser definido é o número de *clusters*. Tipicamente, sua escolha é experimental, onde diversos valores são avaliados.

Deseja-se também determinar os centros $\boldsymbol{\mu}$ dos *clusters*. Uma boa alternativa para isso é o algoritmo de aprendizado não-supervisionado *K-Means*, cujo princípio é formar K conjuntos disjuntos, de tal forma que a distância de cada amostra $x_i^{(k)}$ de um k-ésimo *cluster* pro seu respectivo centro μ_k seja mínima. Os centros, inicializados aleatoriamente, vão sendo ajustados a medida que ocorrem novas designações de amostras aos *clusters*.

```
changes <- TRUE
# Keep iterating until there is no change to the centroids
while(changes){

  Y_last <- Y

  # Assign sample n to the nearest cluster
  for(n in 1:n_samples){
    min_dist <- Inf
    dist <- 0
    for(k in 1:k_clusters){
      if(!all(is.nan(centroid[k,]))){
        dist <- sqrt(sum((X[n,] - centroid[k,])^2))
        if(dist < min_dist){
          min_dist <- dist
          Y[n] <- k
        }
      }
      # Fix empty clusters
      else{
        centroid[k,] <- X[n,]
        Y[n] <- k
      }
    }
  }

  # Update cluster centroids
  for(k in 1:k_clusters){
    index <- which(Y == k)
    cluster[k] <- list(X[index,])
    centroid[k,] <- colMeans(do.call(rbind, cluster[k]))
  }

  # There was no change in the clusters
  if(all(Y_last == Y)){
    changes <- FALSE
  }
}
```

Finalmente, todas as larguras σ das funções radiais podem ser estimadas com sim-

plicidade por (2):

$$\sigma = \frac{d}{\sqrt{2k}} \tag{2}$$

onde d é a distância máxima entre quaisquer dois centros. Vale a pena ressaltar que existem inúmeras outras heurísticas para determinar tal parâmetro, inclusive algumas que atribuem larguras distintas para cada função radial, mas por ora, não há necessidade de metodologias mais elaboradas.

```
  # Heuristic to estimate width of Gaussian kernel
width_estimate <- function(centroid){

  # Number of clusters
  k_clusters <- dim(as.matrix(centroid))[1]

  # Calculate maximum distance between two clusters
  max_dist <- 0
  for(i in 1:k_clusters){
    for(j in 1:k_clusters){
      dist <- norm(as.matrix(centroid[i,] - centroid[j,]))
      if(!is.na(dist) && dist > max_dist){
        max_dist <- dist
      }
    }
  }
  # Calculate width of Gaussian kernel
  sigma = max_dist/sqrt(2*k_clusters)

  return(sigma)
}
```

De maneira similar às *Extreme Learning Machines*, a matriz de pesos $\boldsymbol{W}$, responsável por mapear a camada intermediária à camada de saída, também pode ser obtida diretamente por (3):

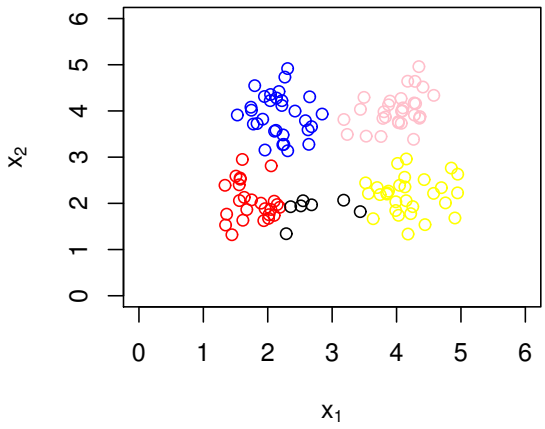
$$\boldsymbol{W} = \boldsymbol{A}_1^{+}\boldsymbol{Y} \tag{3}$$

Assim, a camada de saída de uma rede RBF pode apenas reproduzir o comportamento de uma rede *adaline*, uma vez que os dados recebidos da camada anterior são linearmente separáveis (Braga, 2000).

```
    # Compute weights of the hidden layer (n_neurons x 1)
  W <- pseudoinverse(A_1) %*% Y

  # Mapping from hidden layer to output layer (n_samples x 1)
  Y_hat <- A_1 %*% W
```

Problema 1 - Exclusive OR (XOR) Com a finalidade de avaliar o desempenho das *Radial Basis Function networks* e gerar potenciais comparações com as *Extreme Learning Machines*, os mesmos problemas foram considerados. No caso da classificação do XOR, o número de *clusters* foi determinado empiricamente em $k = 5$ e os centros $\boldsymbol{\mu}$ foram obtidos pelo *K-Means* (4):



$$\boldsymbol{\mu} = \begin{bmatrix} 2.1788 & 3.9139 \\ 1.7594 & 2.0433 \\ 2.7142 & 1.8756 \\ 3.9974 & 4.0205 \\ 4.1920 & 2.1837 \end{bmatrix} \tag{4}$$

Figura 1: Particionamento dos dados em 5 *clusters* pelo *K-Means*.

Por fim, todas as larguras das funções de base radiais foram definidas em $\sigma = 1.3329$ pela heurística apresentada. Percebe-se que a superfície de separação obtida pela *Radial Basis Function network* (Figura 2) se mostra mais suave quando comparada às superficíes geradas pela *Extreme Learning Machine* com 10 e 20 neurônios na camada escondida. Para o mesmo número de neurônios na camada escondida, isto é, 5 neurônios, a ELM sequer conseguiu se ajustar aos dados.

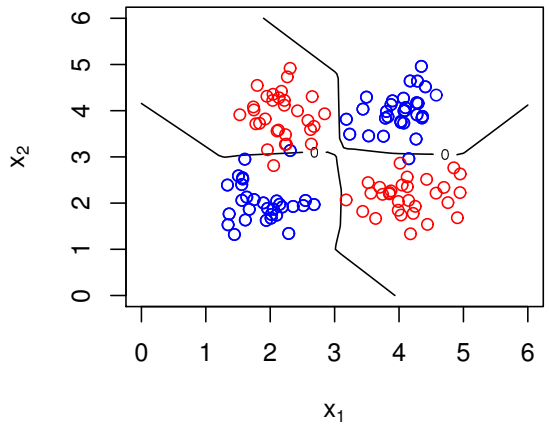


Figura 2: Superfície de separação obtida pela *Radial Basis Function network* com 5 neurônios na camada escondida ($k = 5$).

Problema 2 - Wisconsin Breast Cancer No contexto do *dataset Wisconsin Breast Cancer*, os dados foram divididos de forma similar ao da resolução pela ELM, ou seja, em duas partições aleatórias: conjunto de treinamento e conjunto de teste, sendo o primeiro responsável por conter 70% das amostras e o segundo por conter os 30% restantes. O

processo de treinamento foi realizado com um total de 10 neurônios na camada escondida ($k = 10$), sendo, portanto, um número extraordinariamente menor que o número de neurônios necessários para um bom desempenho das *Extreme Learning Machines* ($k = 105$).

Segundo as métricas de avaliação numéricas, além de exigir menor complexidade em quantidade de neurônios, a rede RBF se apresentou substancialmente mais precisa que a ELM. Em pontos percentuais, a sensibilidade foi superior em 2.53, a especificidade foi superior em 1.59 e o erro quadrático médio foi inferior em 6.23 (Tabela 1).

threshold	Sensibilidade	Especificidade	Erro Quadrático Médio
0.5	0.9873	0.9920	2.17×10^{-2}

Tabela 1: Métricas de avaliação de desempenho.

Diante da matriz de confusão, é possível ratificar com ainda mais convicção o bom desempenho das *Radial Basis Function networks* para esse problema. O percentual de classificações corretas foi superior a 98%, visto que apenas 2 de 205 pacientes foram diagnosticados incorretamente (Tabela 2).

		Actual	
		M	B
Predicted	M	125	1
	B	1	78

Tabela 2: Matriz de confusão.

Conclusão

À medida que a complexidade dos problemas cresce, torna-se compreensível a necessidade de topologias mais avançadas de redes neurais artificiais. Ainda que extremamente simples quando comparadas às arquiteturas mais atuais, redes do tipo *feed-forward*, como *Radial Basis Function networks*, já introduzem uma camada intermediária entre as camadas de entrada e saída, dando princípio, posteriormente, a modelos mais elaborados de múltiplas camadas.

Referências

Braga, A. Redes neurais artificiais: teoria e aplicações. Livros Técnicos e Científicos, 2000.

David S. Broomhead; David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. 1988.