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

Exercício Computacional 3: Extreme Learning Machine

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] o algoritmo de treinamento de redes neurais *feedforward*: *Extreme Learning Machine (ELM)*. Tal algoritmo foi capaz de treinar modelos com 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.

Extreme Learning Machines (ELMs)

Propostas por Guang-Bin Huang, Qin-Yu Zhu e Chee-Kheong Siew (2004), as *Extreme Learning Machines* surgiram com o intuito de apresentar uma metodologia alternativa ao treinamento iterativo de redes neurais por método do gradiente.

O treinamento das ELMs é realizado de forma analítica, onde os pesos da camada de entrada $\boldsymbol{W_1}$ são constantes e gerados aleatoriamente de uma distribuição uniforme $\mathcal{U}(-0.5, 0.5)$. Além disso, sua arquitetura é composta por uma camada intermediária, cuja entrada é mapeada pela matriz $\boldsymbol{A_1}$ (1):

$$\boldsymbol{A_1} = \psi(\boldsymbol{X}\boldsymbol{W_1}) \tag{1}$$

onde $\psi(.)$ é uma função de ativação, sendo tipicamente utilizadas as funções sigmóide e tangente hiperbólica.

```
# Random weights of input layer (n_features x n_neurons)
W_1 <- matrix(runif(n_features, min = -0.5, max = 0.5),
                    nrow = n_features, ncol = n_neurons)
W_1 <- replicate(n_neurons, runif(n_features, -0.5, 0.5))

# Mapping from input layer to hidden layer (n_samples x n_neurons)
A_1 <- tanh(X %*% W_1)
```

A matriz de pesos $\boldsymbol{W_2}$, responsável por mapear a camada intermediária à camada de saída, pode ser expressa por uma simplificação linear da equação $\Phi(\boldsymbol{A_1}\boldsymbol{W_2}) = \boldsymbol{Y}$, sendo, portanto, obtida diretamente por (2):

$$\boldsymbol{W_2} = \boldsymbol{A_1^+}\boldsymbol{Y} \tag{2}$$

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

onde $\boldsymbol{A_1^+}$ é a pseudo-inversa de $\boldsymbol{A_1}$. Por fim, uma simples combinação linear dos pesos $\boldsymbol{W_2}$ com a matriz de mapeamento $\boldsymbol{A_1}$ permite encontrar as predições (3):

$$\hat{\boldsymbol{Y}} = \boldsymbol{A_1 W_2} \tag{3}$$

```
  # Compute weights of the hidden layer (n_neurons x 1)
W_2 <- pseudoinverse(A_1) %*% Y

# Mapping from hidden layer to output layer (n_samples x 1)
Y_hat <- A_1 %*% W_2
```

Problema 1 - Exclusive OR (XOR) Através dos Exercícios Computacionais 1 e 2, percebe-se a eficácia da rede *perceptron* simples para problemas em que a separabilidade linear dos dados é garantida. No entanto, a maioria dos problemas reais não sustentam tal premissa. O operador XOR, por exemplo, não é linearamente separável, uma vez que não existe uma reta capaz de separar as regiões de 0 e 1.

Com o intuito de verificar a robustez das *Extreme Learning Machines* para esse caso, especificadamente, dois conjuntos foram carregados do arquivo *data2classXOR.txt* conforme o problema de classificação do XOR (Figura 1).

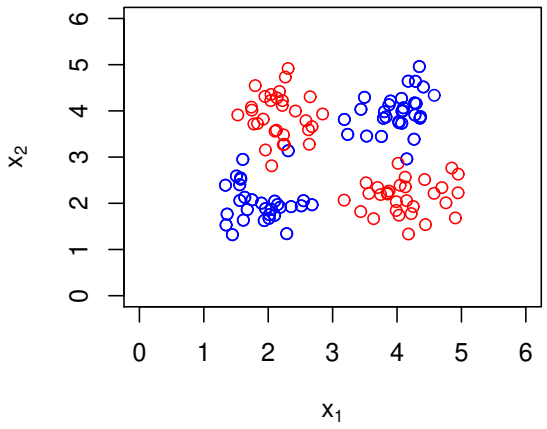

Figura 1: Problema de classificação do ou-exclusivo (XOR).

A definição de hiperparâmetros é uma etapa fundamental para se obter um bom desempenho com algoritmos de aprendizado de máquina. Assim como a taxa de aprendizado η das redes *perceptron* e *adaline*, as ELMs possuem um hiperparâmetro que deve ser definido com bastante cautela: o número de neurônios na camada escondida. Quando o número é substancialmente pequeno, o modelo pode não se adaptar aos dados de treinamento (*underfitting*). E quando muito grande, o modelo pode ficar extremamente especialista, ou seja, adaptado apenas para os dados de treinamento, sem generalizar dados novos (*overfitting*).

Por consequência, o número de neurônios na camada escondida foi definido empiricamente, visando um bom ajuste visual da superfície de separação nos dados. Testes

com poucos neurônios promoveram *underfitting* (Figura 2) e testes com muitos neurônios promoveram *overfitting* (Figura 2), conforme o previsto.

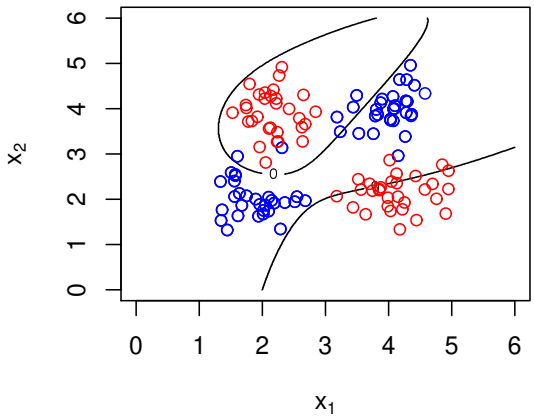

Figura 2: Superfície de separação obtida pela *Extreme Learning Machine* com 5 neurônios na camada escondida.

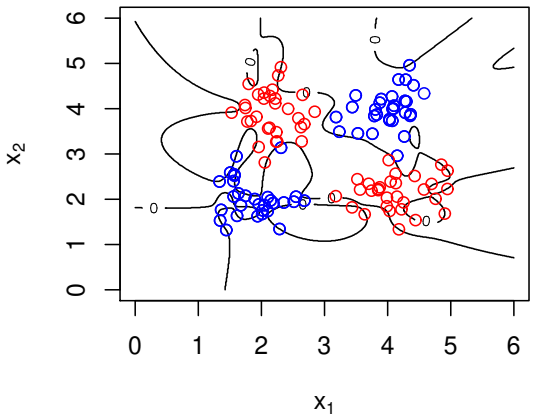

Figura 3: Superfície de separação obtida pela *Extreme Learning Machine* com 100 neurônios na camada escondida.

Já arquiteturas com cerca de 10 a 20 neurônios na camada escondida promoveram superfícies de separações bastante convincentes.

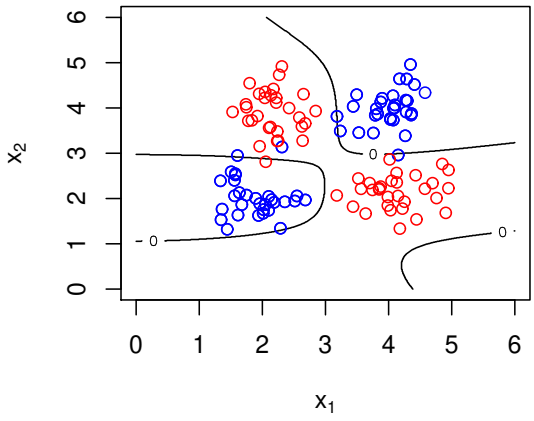

Figura 4: Superfície de separação obtida pela *Extreme Learning Machine* com 10 neurônios na camada escondida.

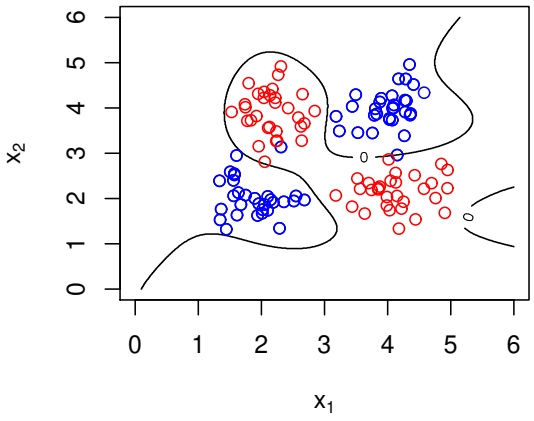

Figura 5: Superfície de separação obtida pela *Extreme Learning Machine* com 20 neurônios na camada escondida.

Problema 2 - Wisconsin Breast Cancer A fim de avaliar o desempenho do modelo também para dados reais, o *dataset Wisconsin Breast Cancer* foi considerado como *benchmark*. Ele compreende 9 *features* calculadas a partir de imagens dos núcleos celulares presentes na mama de 683 pacientes. O objetivo é utilizar as *Extreme Learning Machines* para classificar os casos dos pacientes em "M"(*malignant*) e "B"(*benign*).

Para tal fim, os dados foram divididos 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 105 neurônios na camada escondida. Apesar de ser um valor bem alto quando comparado com o problema do XOR, a escolha se justifica, uma vez que o problema de classificação de tumores na mama é maior e mais complexo.

```
# Splitting data for training and test
proportion <- 0.7
set.seed(10)
perm <- sample(dim(X)[1])
train_index <- perm[1:round(proportion*n_samples)]
test_index <- perm[round(proportion*n_samples + 1):n_samples]
X_train <- X[train_index,]
Y_train <- Y[train_index]
X_test <- X[test_index,]
Y_test <- Y[test_index]
```

Uma vez que os dados estão no plano $\mathbb{R}^9$, a visualização da superfície de separação se torna inviável para avaliar o desempenho do modelo. Assim, métricas de avaliação numéricas, como sensibilidade, especificidade e erro devem ser consideradas (Tabela 1).

threshold	Sensibilidade	Especificidade	Erro Quadrático Médio
0.5	0.9620	0.9761	8.4×10^{-2}

Tabela 1: Métricas de avaliação de desempenho.

A matriz de confusão também é uma boa alternativa para casos com dimensões maiores, apresentando os números de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos (Tabela 2).

		Actual	
		M	B
Predicted	M	123	3
	B	3	76

Tabela 2: Matriz de confusão.

Diante dos valores apresentados, pode-se afirmar que o desempenho das *Extreme Learning Machines* para esse problema foi bastante satisfatório. Tanto o percentual de classificações corretas para os que possuem o tumor maligno (especificidade) quanto para aqueles que possuem o tumor benigno (sensibilidade) foram acima de 96%. Além disso, o erro quadrático médio calculado foi pequeno, visto que apenas 6 pacientes de 205 (conjunto de teste) foram classificados incorretamente.

Conclusão

À medida que a complexidade dos problemas cresce, torna-se compreensível a necessidade de topologias mais avançadas de redes neurais artificiais. Ainda que ex-

tremamente simples quando comparadas às arquiteturas mais atuais, redes do tipo *single-layer perceptron* já introduzem uma camada intermediária entre as camadas de entrada e saída, dando princípio, posteriormente, a modelos mais elaborados de múltiplas camadas.

Referências

Guang-Bin Huang; Qin-Yu Zhu; Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, v. 2, p. 985–990 vol.2, Julho 2004. ISSN 1098-7576.