

## Exercício

---

Frederico Coelho

October 3, 2019

### BACKPROPAGATION

Perceptrons de múltiplas camadas (MLP) são capazes de aproximar qualquer função quando possuem duas camadas escondidas, quando há apenas uma camada intermediária é possível a aproximação qualquer função contínua. O algoritmo mais difundido para o treinamento de uma rede MLP é chamado *backpropagation*, ele utiliza o conceito da retro propagação do erro da rede para corrigir os pesos da última camada até a primeira. O ajuste dos pesos na camada de saída é dado por

$$\Delta w_{ji} = \alpha e_j(n) f'(u_j(n)) h_i(n)$$

em que  $i$  é o índice dos neurônios da camada escondida,  $j$  é o índice dos neurônios da camada de saída,  $e_j(n)$  é o erro apresentado na saída para a entrada  $n$ ,  $h_i(n)$  é a saída do  $i$ -ésimo neurônio da camada escondida,  $u(n)$  é o somatório das saídas da camada escondida ponderada pelos pesos  $w_{jk}$  e  $f'$  é a derivada da função de ativação da camada de saída. A constante  $\alpha$  é uma taxa de aprendizado arbitrária que, em geral, deve apresentar um valor baixo.

Para a camada escondida deve ser considerada a retro propagação do erro, assim o ajuste dos pesos da camada escondida é dado por:

$$\Delta w_{ik} = \eta h'_i(u_i(n)) \sum_j e_j(n) f'(u_j(n)) w_{ji}(n) x_k(n)$$

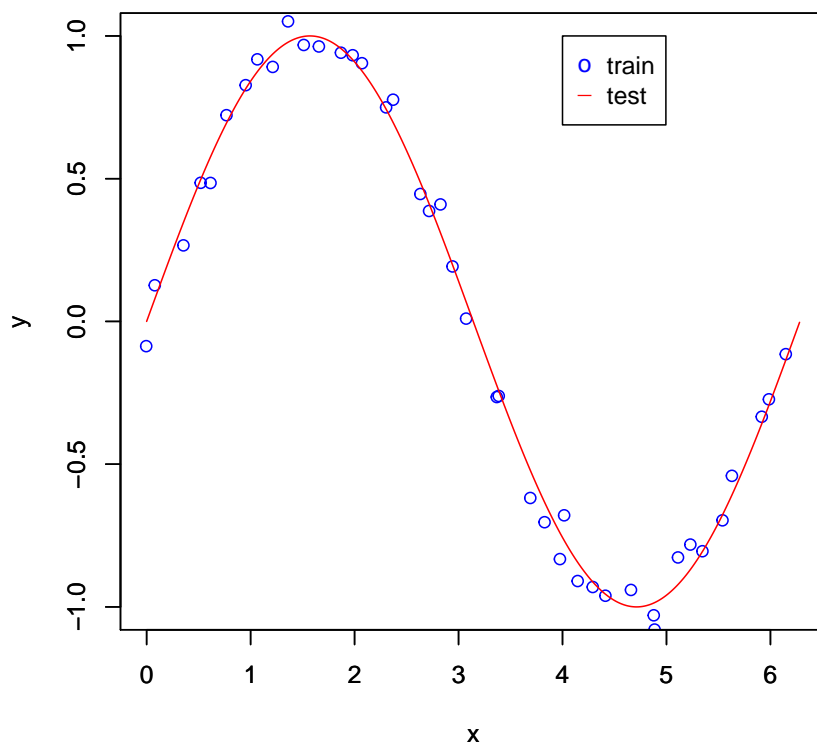
em que  $k$  é o índice da entrada  $x$ ,  $h'$  é a derivada da função de ativação da camada escondida, no caso da função linear a derivada da função é uma constante. E  $\eta$  é a taxa de aprendizado.

## INSTRUÇÕES

Para observar que o MLP é capaz de aproximar qualquer função contínua, o aluno deve realizar a regressão de um ciclo de uma senoide com uma MLP usando backpropagation. A função de ativação da camada de saída deve ser linear, e devem haver 3 neurônios na camada escondida com função de ativação tangente hiperbólica. Implementar o backpropagation.

1. Criar os dados de treinamento e de teste, de acordo com uma senoidal acrescida de um erro:

```
> x_train<-seq(from=0, to=2*pi, by =0.15)
> x_train<-x_train + (runif(length(x_train))-0.5)/5
> i <- sample(length(x_train))
> x_train <- x_train[i]
> y_train <- sin(x_train)
> y_train<-y_train + (runif(length(y_train))-0.5)/5
> plot(x_train,y_train,col='blue',xlim = c(0,2*pi),
+      ylim = c(-1,1),xlab = 'x',ylab = 'y')
> x_test <-seq(from=0, to=2*pi, by =0.01)
> y_test <-sin(x_test)
> par(new=T)
> plot(x_test,y_test,col='red',type='l',xlim = c(0,2*pi),
+      ylim = c(-1,1),xlab = 'x',ylab = 'y')
> legend(x=4, y=1, legend = c('train','test'),
+      col = c('blue','red'),pch=c('o','_'))
```



2. Definir uma RNA com uma camada escondida de 3 neurônios com função de ativação tangente hiperbólica e um neurônio de saída com função de ativação linear. De forma semelhante ao explicado em sala de aula (vide slides).
3. Implementar um função de treinamento utilizando o backpropagation e ajustar um modelo com os dados de treinamento.
4. Testar o modelo obtido com os dados de teste. (separe 20% dos dados aleatoriamente para teste).
5. Plotar as saídas da função aproximada e os valores esperados y.

O aluno deverá entregar um relatório em pdf contendo os gráficos pedidos e suas conclusões.

Exercício original de Antônio Braga