

Exercício 9.

Nome: Lucas Kou Kinoshita

RM: 2019021557

data: 31/05/2025

Implementação:

Primeiramente, os conjuntos sintéticos de treinamento são criados como sugerido no enunciado do exercício, com 45 amostras com valores de x amostrados entre 0 e 2π e valores de $y = \text{seno}(x) + \text{ruído}$ (uniformemente amostrado no intervalo $[-0.1, 0.1]$). Em seguida, criam-se os conjuntos sintéticos de teste, composto de valores de x entre 0 e 2π , obtidos com passo $\delta = 0.01$, e $y = \text{seno}(x)$.

```
x_range <- c(0, 2 * pi)
N_train <- 45
noise_range <- 0.1 # noise uniformly sampled in [-0.1, 0.1]
xtr <- runif(N_train, min = x_range[1], max = x_range[2])
ytr <- sin(xtr) + runif(N_train, min = -noise_range, max = noise_range)

xtst <- seq(0, 2 * pi, 0.01)
ytst <- sin(xtst)

plot(x = xtr, y = ytr, col = 'blue', pch = 1, xlab = "x", ylab = "y",
     main = "Training and Test Data", xaxt = "n",
     xlim = x_range, ylim = c(-1.5, 1.5))
axis(side = 1, at = seq(0, 2*pi, pi/2),
     labels = c("0", "pi/2", "pi", "3pi/2", "2pi"))
lines(x = xtst, y = ytst, type = 'line', col = 'red')
```

Figura 1: Inicialização dos conjuntos de dados

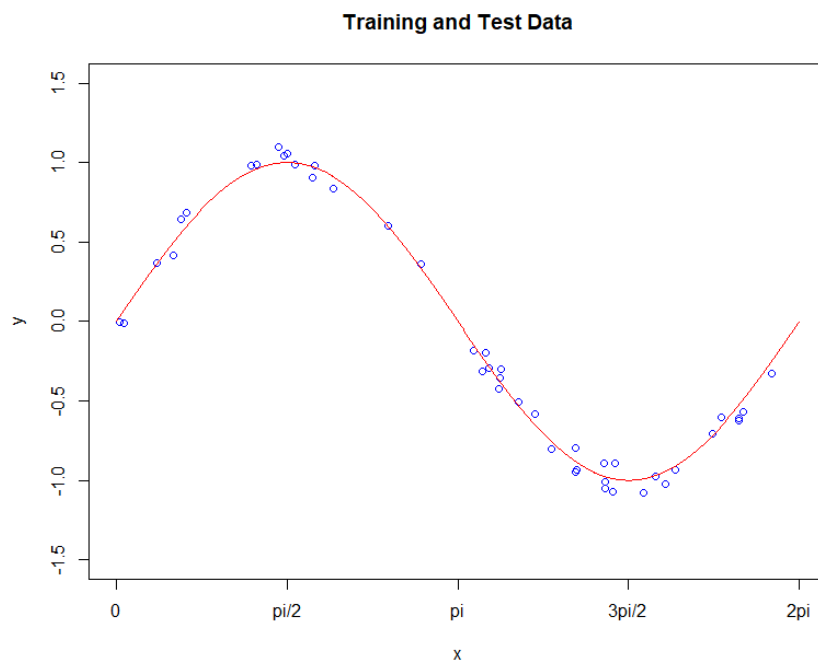


Figura 2: Visualização dos conjuntos de dados

Em seguida, o algoritmo de treinamento de uma MLP elaborado em sala é adaptado para uma rede com entrada e saída singular, contendo três neurônios na camada escondida e ativação linear na saída como observado abaixo:

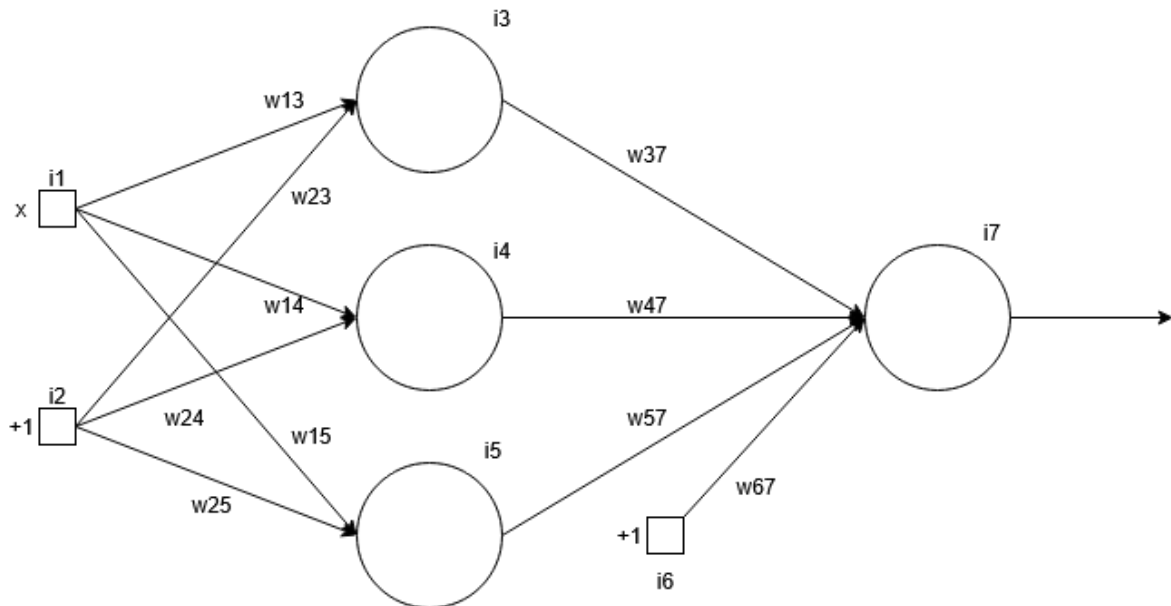


Figura 3: Visualização da rede

Como são pedidas cinco inicializações diferentes da MLP, após a criação dos conjuntos de dados, um loop de 5 inicializações foi criado:

```
n_initializations <- 5
mse_results <- numeric(n_initializations)
final_yhat_test_for_plot <- NULL

for (run in 1:n_initializations) {
  cat(paste0("\n--- Starting Run ", run, " ---\n"))

  # Entradas camada oculta
  w13 <- runif(1) - 0.5
  w23 <- runif(1) - 0.5

  w14 <- runif(1) - 0.5
  w24 <- runif(1) - 0.5

  w15 <- runif(1) - 0.5
  w25 <- runif(1) - 0.5

  # Saídas da camada oculta
  w37 <- runif(1) - 0.5
  w47 <- runif(1) - 0.5
  w57 <- runif(1) - 0.5
  w67 <- runif(1) - 0.5

  maxepocas <- 5000
  nepocas <- 0
  tol <- 0.00001
  eepoca <- tol + 1
  evec <- numeric(maxepocas)
  eta <- 0.01 # passo

  # Bias
  i2 <- +1
  i6 <- +1
}
```

Figura 4: Início de loop de experimento

Dentro deste último loop, foi implementado um segundo loop para o processo de treinamento iterativo da MLP:

```
while ((nepocas < maxepocas) && (eepoca > tol)) {  
  ei2 <- 0 # erro quadrático acumulado por época  
  
  xseq <- sample(N_train)  
  for (i in 1:N_train) {  
    irand <- xseq[i]  
    input_x <- xtr[irand]  
    y7 <- ytr[irand]  
  
    # Forward Pass  
    # Camada escondida  
    u3 <- input_x * w13 + i2 * w23  
    i3 <- tanh(u3)  
  
    u4 <- input_x * w14 + i2 * w24  
    i4 <- tanh(u4)  
  
    u5 <- input_x * w15 + i2 * w25  
    i5 <- tanh(u5)  
  
    # Camada de saída  
    u7 <- i3 * w37 + i4 * w47 + i5 * w57 + i6 * w67  
    i7 <- u7 # linear sem usar tanh  
  
    # Calcular erro  
    e7 <- y7 - i7  
    ei2 <- ei2 + (e7^2) / 2  
  
    # Cálculo da variação dos pesos da camada de saída  
    d7 <- e7*sech2(u7)
```

Figura 5: Entradas e saídas da camada oculta e de saída

```
# Cálculo da variação dos pesos da camada de saída  
d7 <- e7*sech2(u7)  
  
dw37 <- eta * d7 * i3  
dw47 <- eta * d7 * i4  
dw57 <- eta * d7 * i5  
dw67 <- eta * d7 * i6  
  
# Cálculo da variação dos pesos da camada intermediária  
e3 <- d7 * w37  
e4 <- d7 * w47  
e5 <- d7 * w57  
  
d3 <- e3 * sech2(u3)  
d4 <- e4 * sech2(u4)  
d5 <- e5 * sech2(u5)  
  
# Atualização dos pesos da camada oculta  
dw13 <- eta * d3 * i1  
dw23 <- eta * d3 * i2  
  
dw14 <- eta * d4 * i1  
dw24 <- eta * d4 * i2  
  
dw15 <- eta * d5 * i1  
dw25 <- eta * d5 * i2
```

Figura 6: Cálculo das variações dos pesos

```

# Atualização de todos os pesos
# Camada oculta
w13 <- w13 + dw13
w23 <- w23 + dw23

w14 <- w14 + dw14
w24 <- w24 + dw24

w15 <- w15 + dw15
w25 <- w25 + dw25

# Camada de saída
w37 <- w37 + dw37
w47 <- w47 + dw47
w57 <- w57 + dw57
w67 <- w67 + dw67

```

Figura 7: Atualização dos pesos

```

nepocas <- nepocas + 1
evec[nepocas] <- ei2 / N_train # erro quadrático por amostra
eepoca <- evec[nepocas]

```

Figura 8: Erro quadrático por amostra e atualização da época

Por fim, os loop de treinamento é encerrado e o cálculo do MSE é realizado antes do encerramento do loop de inicialização.

```

# --- MSE no conjunto de testes ---
yhat_tst <- numeric(length(xtst))
for (k in 1:length(xtst)) {
  i1_test <- xtst[k]

  i3_test <- tanh(i1_test * w13 + i2 * w23)
  i4_test <- tanh(i1_test * w14 + i2 * w24)
  i5_test <- tanh(i1_test * w15 + i2 * w25)

  yhat_tst[k] <- i3_test * w37 + i4_test * w47 + i5_test * w57 + i6 * w67
}

mse_test <- mean((ytst - yhat_tst)^2)
mse_results[run] <- mse_test
cat(paste0("MSE for Run ", run, ": ", round(mse_test, 6), "\n"))

if (run == n_initializations) {
  final_yhat_test_for_plot <- yhat_tst
}
}

```

Figura 9: Fim de loop de experimento

Por último, são obtidos os dados gerais de cada experimento e a aproximação realizada pelo último experimento é plotada para visualização da qualidade alcançada pela MLP

```

# --- Resultados ---
cat("\n--- MSE Results Across All Runs ---\n")
cat("Individual MSEs:", round(mse_results, 6), "\n")
cat("Mean MSE:", round(mean(mse_results), 6), "\n")
cat("Standard Deviation of MSE:", round(sd(mse_results), 6), "\n")

plot(x = xtr, y = ytr, col = 'blue', pch = 1, xlab = "x", ylab = "y",
     main = "MLP Sine Approximation (One Run Example)",
     xlim = x_range, ylim = c(-1.5, 1.5), xaxt = "n")
axis(side = 1, at = seq(0, 2*pi, pi/2),
     labels = c("0", "pi/2", "pi", "3pi/2", "2pi"))
lines(x = xtst, y = ytst, col = 'red', lwd = 2) # True sine wave
lines(x = xtst, y = final_yhat_test_for_plot, col = 'black', lty = 2, lwd = 2) # MLP prediction

legend("topright", legend = c("Training Data", "True Sine Wave", "MLP Prediction"),
     col = c("blue", "red", "black"), pch = c(1, NA, NA),
     lty = c(NA, 1, 2), lwd = c(NA, 2, 2))

```

Figura 10: Implementação do log de resultados e do plot

```

--- Starting Run 1 ---
MSE for Run 1: 0.037765

--- Starting Run 2 ---
MSE for Run 2: 0.033662

--- Starting Run 3 ---
MSE for Run 3: 0.011687

--- Starting Run 4 ---
MSE for Run 4: 0.001765

--- Starting Run 5 ---
MSE for Run 5: 0.002674
>
> # --- Resultados ---
> cat("\n--- MSE Results Across All Runs ---\n")

--- MSE Results Across All Runs ---
> cat("Individual MSEs:", round(mse_results, 6), "\n")
Individual MSEs: 0.037765 0.033662 0.011687 0.001765 0.002674
> cat("Mean MSE:", round(mean(mse_results), 6), "\n")
Mean MSE: 0.017511
> cat("Standard Deviation of MSE:", round(sd(mse_results), 6), "\n")
Standard Deviation of MSE: 0.017125
>

```

Figura 11: Log de resultados obtidos

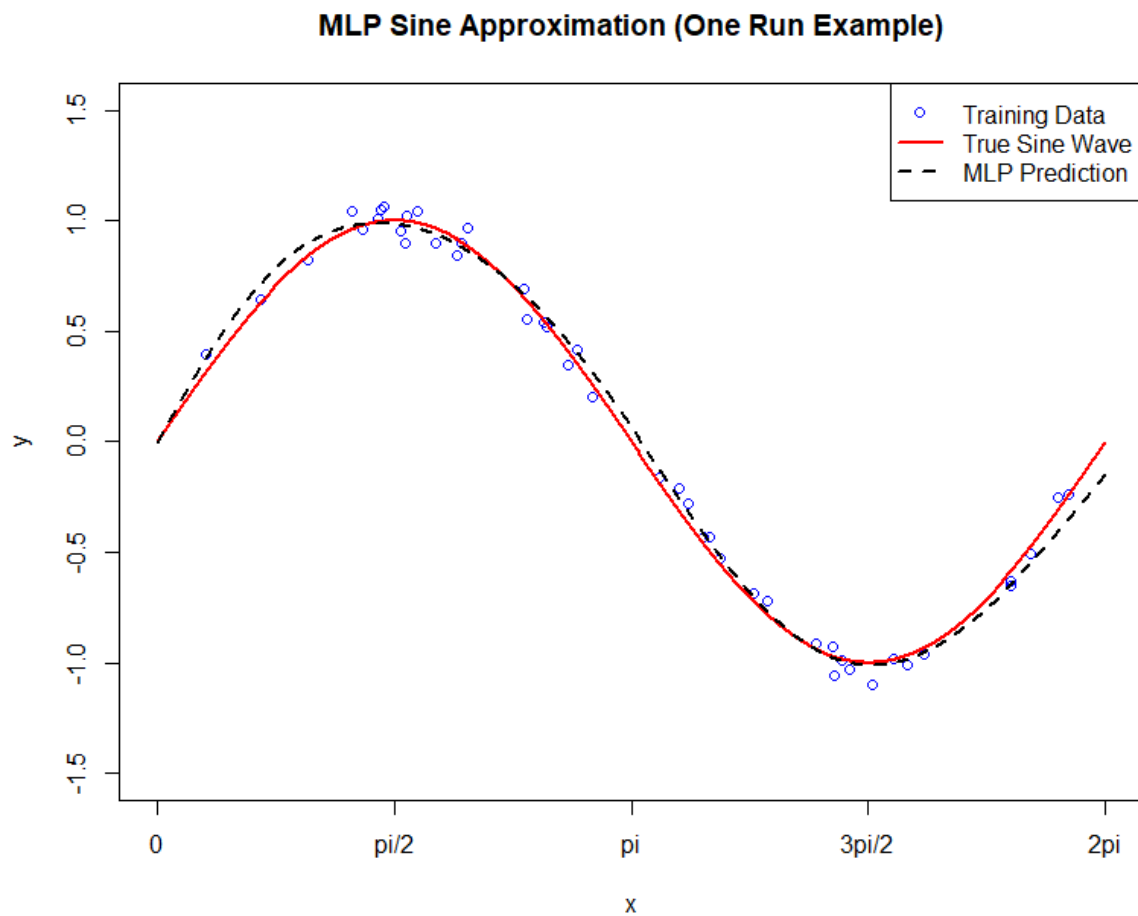


Figura 12: Visualização da aproximação alcançada pela solução da MLP

Conclusão:

Através da inspeção visual da senoide obtida pela aproximação alcançada pela MLP e pelos baixos valores de erro quadrático médio, podemos concluir que o exercício foi realizado com sucesso.