Artificial neural network architecture

Simultaneous mode

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Split-percentage | Hidden\_1 | Hidden\_2 | Hidden\_3 | Training error | Steps |  | Output parameters | | | | | | |
| Accuracy analysis | 1 | 2 | 3 | 4 | 5 | 6 | Average |
| 80%-20% | 6 |  |  | 1.5473 | 4754 | MAE | 0.0000 | 0.0423 | 0.0000 | 0.0000 | 0.0282 | 0.1408 | 0.0352 |
| 80%-20% | 6 |  |  | 1.5473 | 4754 | RMSE | 0.0000 | 0.2056 | 0.0000 | 0.0000 | 0.1678 | 0.3753 | 0.1248 |
| 80%-20% | 6 |  |  | 1.5473 | 4754 | uMbRAE | 0.1360 | 0.3148 | 0.1181 | 0.1181 | 0.1736 | 0.8205 | 0.2802 |
| 60%-40% | 7 |  |  | 1.1266 | 5358 | MAE | 0.0084 | 0.0840 | 0.0168 | 0.0168 | 0.0588 | 0.1345 | 0.0532 |
| 60%-40% | 7 |  |  | 1.1266 | 5358 | RMSE | 0.0917 | 0.2899 | 0.1296 | 0.1296 | 0.2425 | 0.3667 | 0.2083 |
| 60%-40% | 7 |  |  | 1.1266 | 5358 | uMbRAE | 0.1498 | 0.3296 | 0.1333 | 0.1388 | 0.1841 | 0.8030 | 0.2898 |
| 60%-40% | 5 | 5 |  | 1.0815 | 7575 | MAE | 0.0084 | 0.0756 | 0.0084 | 0.0084 | 0.0504 | 0.1429 | 0.0490 |
| 60%-40% | 5 | 5 |  | 1.0815 | 7575 | RMSE | 0.0917 | 0.2750 | 0.0917 | 0.0917 | 0.2245 | 0.3780 | 0.1921 |
| 60%-40% | 5 | 5 |  | 1.0815 | 7575 | uMbRAE | 0.1498 | 0.3446 | 0.1226 | 0.1280 | 0.1841 | 0.8030 | 0.2887 |
| 60%-40% | 6 | 6 |  | 1.2098 | 3836 | MAE | 0.0000 | 0.0588 | 0.0000 | 0.0084 | 0.0420 | 0.1176 | 0.0378 |
| 60%-40% | 6 | 6 |  | 1.2098 | 3836 | RMSE | 0.0000 | 0.2425 | 0.0000 | 0.0917 | 0.2050 | 0.3430 | 0.1470 |
| 60%-40% | 6 | 6 |  | 1.1643 | 7430 | uMbRAE | 0.1515 | 0.3475 | 0.1176 | 0.1515 | 0.1728 | 0.9192 | 0.3100 |
| 60%-40% | 6 | 6 | 6 | 1.0529 | 3014 | MAE | 0.0000 | 0.0756 | 0.0084 | 0.0000 | 0.0420 | 0.1513 | 0.0462 |
| 60%-40% | 6 | 6 | 6 | 1.0529 | 3014 | RMSE | 0.0000 | 0.2750 | 0.0917 | 0.0000 | 0.2050 | 0.3889 | 0.1601 |
| 60%-40% | 6 | 6 | 6 | 1.0529 | 3014 | uMbRAE | 0.1388 | 0.3446 | 0.1333 | 0.1280 | 0.1724 | 0.7761 | 0.2822 |
| 60%-40% | 6 | 6 | 4 | 1.1552 | 5105 | MAE | 0.0000 | 0.0840 | 0.0084 | 0.0084 | 0.0420 | 0.1261 | 0.0448 |
| 60%-40% | 6 | 6 | 4 | 1.1552 | 5105 | RMSE | 0.0000 | 0.2899 | 0.0917 | 0.0917 | 0.2050 | 0.3550 | 0.1722 |
| 60%-40% | 6 | 6 | 4 | 1.1552 | 5105 | uMbRAE | 0.1388 | 0.3600 | 0.1226 | 0.1280 | 0.1610 | 0.8030 | 0.2856 |
| 70%-30% | 6 | 6 | 6 | 1.2095 | 6262 | MAE | 0.0000 | 0.0737 | 0.0000 | 0.0105 | 0.0526 | 0.1895 | 0.0544 |
| 70%-30% | 6 | 6 | 6 | 1.2095 | 6262 | RMSE | 0.0000 | 0.2714 | 0.0000 | 0.1026 | 0.2294 | 0.4353 | 0.1731 |
| 70%-30% | 6 | 6 | 6 | 1.2095 | 6262 | uMbRAE | 0.1377 | 0.3669 | 0.1176 | 0.1243 | 0.1728 | 0.8447 | 0.2940 |
| 70%-30% | 6 | 6 | 7 | 1.3098 | 4899 | MAE | 0.0211 | 0.0632 | 0.0000 | 0.0211 | 0.0632 | 0.1158 | 0.0526 |
| 70%-30% | 6 | 6 | 7 | 1.3098 | 4899 | RMSE | 0.1451 | 0.2513 | 0.0000 | 0.1451 | 0.2513 | 0.3403 | 0.1976 |
| 70%-30% | 5 | 5 | 6 | 1.170 | 8327 | uMbRAE | 0.152 | 0.348 | 0.131 | 0.124 | 0.159 | 0.810 | 0.364 |
| 80%-20% | 6 | 6 | 7 | 1.435 | 5150 | MAE | 0.000 | 0.056 | 0.000 | 0.000 | 0.042 | 0.141 | 0.061 |
| 80%-20% | 6 | 6 | 7 | 1.435 | 5150 | RMSE | 0.000 | 0.237 | 0.000 | 0.000 | 0.206 | 0.375 | 0.194 |
| 80%-20% | 6 | 6 | 7 | 1.435 | 5150 | uMbRAE | 0.136 | 0.315 | 0.118 | 0.118 | 0.174 | 0.821 | 0.371 |
| 80%-20% | 6 | 6 | 4 | 1.418 | 6310 | MAE | 0.000 | 0.070 | 0.000 | 0.000 | 0.042 | 0.141 | 0.061 |
| 80%-20% | 6 | 6 | 4 | 1.418 | 6310 | RMSE | 0.000 | 0.265 | 0.000 | 0.000 | 0.206 | 0.375 | 0.194 |
| 80%-20% | 6 | 6 | 4 | 1.418 | 6310 | uMbRAE | 0.136 | 0.365 | 0.118 | 0.118 | 0.174 | 0.775 | 0.356 |

Air mode

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Split-percentage | Hidden\_1 | Hidden\_2 | Hidden\_3 | Training error | Steps |  | Output parameters | | | | | |
| Accuracy analysis | 1 | 2 | 3 | 4 | 6 | Average |
| 60%-40% | 6 |  |  | 0.07194 | 4392 | MAE | 0.0000 | 0.0000 | 0.0078 | 0.0078 | 0.0078 | 0.0047 |
| 60%-40% | 6 |  |  | 0.07194 | 4392 | RMSE | 0.0000 | 0.0000 | 0.0884 | 0.0884 | 0.0884 | 0.0530 |
| 60%-40% | 6 |  |  | 0.07194 | 4392 | uMbRAE | 0.1327 | 0.1130 | 0.1378 | 0.1584 | 0.8028 | 0.2690 |
| **80%-20%** | **4** | **5** |  | **0.05225** | **4352** | MAE | **0.0156** | **0.0000** | **0.0000** | **0.0000** | **0.0000** | **0.0031** |
| 80%-20% | 4 | 5 |  | 0.05225 | 4352 | RMSE | 0.1250 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0250 |
| 80%-20% | 4 | 5 |  | 0.05225 | 4352 | uMbRAE | 0.1429 | 0.1034 | 0.1327 | 0.1429 | 0.7778 | 0.2599 |
| 80%-20% | 4 | 5 | 3 | 0.10445 | 2528 | MAE | 0.0000 | 0.0313 | 0.0156 | 0.0000 | 0.0313 | 0.0156 |
| 80%-20% | 4 | 5 | 3 | 0.10445 | 2528 | RMSE | 0.0000 | 0.1768 | 0.1250 | 0.0000 | 0.1768 | 0.0957 |
| 80%-20% | 4 | 5 | 3 | 0.10445 | 2528 | uMbRAE | 0.1228 | 0.1429 | 0.1327 | 0.1429 | 0.7778 | 0.2638 |
| 80%-20% | 4 | 5 | 7 | 0.11028 | 1630 | MAE | 0.0000 | 0.0313 | 0.0156 | 0.0000 | 0.0156 | 0.0125 |
| 80%-20% | 4 | 5 | 7 | 0.11028 | 1630 | RMSE | 0.0000 | 0.1768 | 0.1250 | 0.0000 | 0.1250 | 0.0854 |
| 80%-20% | 4 | 5 | 7 | 0.11028 | 1630 | uMbRAE | 0.1228 | 0.1429 | 0.1327 | 0.1429 | 0.8286 | 0.2740 |
| 70%-30% | 4 | 5 | 7 | 0.09286 | 1337 | **MAE** | 0.0000 | 0.0208 | 0.0104 | 0.0000 | 0.0208 | 0.0104 |
| 70%-30% | 4 | 5 | 7 | 0.09286 | 1337 | RMSE | 0.0000 | 0.1443 | 0.1021 | 0.0000 | 0.1443 | 0.0781 |
| 70%-30% | 4 | 5 | 7 | 0.09286 | 1337 | uMbRAE | 0.1294 | 0.1361 | 0.1361 | 0.1429 | 0.8113 | 0.2712 |
| 70%-30% | 4 | 5 | 5 | 0.10807 | 4627 | MAE | 0.0104 | 0.0208 | 0.0000 | 0.0104 | 0.0104 | 0.0104 |
| 70%-30% | 4 | 5 | 5 | 0.10807 | 4627 | RMSE | 0.1021 | 0.1443 | 0.0000 | 0.1021 | 0.1021 | 0.0901 |
| 70%-30% | 4 | 5 | 5 | 0.10807 | 4627 | uMbRAE | 0.1429 | 0.1361 | 0.1361 | 0.1566 | 0.7778 | 0.2699 |
| 60%-40% | 4 | 5 | 5 | 0.07466 | 3057 | MAE | 0.0000 | 0.0156 | 0.0078 | 0.0000 | 0.0156 | 0.0078 |
| 60%-40% | 4 | 5 | 5 | 0.07466 | 3057 | RMSE | 0.0000 | 0.1250 | 0.0884 | 0.0000 | 0.1250 | 0.0677 |
| 60%-40% | 4 | 5 | 5 | 0.07466 | 3057 | uMbRAE | 0.1327 | 0.1327 | 0.1378 | 0.1480 | 0.8028 | 0.2708 |
| 60%-40% | 4 | 5 | 6 | 0.08254 | 3043 | MAE | 0.0078 | 0.0156 | 0.0078 | 0.0000 | 0.0000 | 0.0063 |
| 60%-40% | 4 | 5 | 6 | 0.08254 | 3043 | RMSE | 0.0884 | 0.1250 | 0.0884 | 0.0000 | 0.0000 | 0.0604 |
| 60%-40% | 4 | 5 | 6 | 0.08254 | 3043 | uMbRAE | 0.1429 | 0.1327 | 0.1378 | 0.1480 | 0.7778 | 0.2678 |

Water mode

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Split-percentage | Hidden\_1 | Hidden\_2 | Hidden\_3 | Training error | Steps |  | Output parameters | | | | | |
| Accuracy analysis | 1 | 2 | 3 | 5 | 6 | Average |
| 60%-40% | 6 |  |  | 0.0829 | 1442 | MAE | 0.0175 | 0.0175 | 0.0175 | 0.0088 | 0.0088 | 0.0140 |
| 60%-40% | 6 |  |  | 0.0829 | 1442 | RMSE | 0.1325 | 0.1325 | 0.1325 | 0.0937 | 0.0937 | 0.1169 |
| 60%-40% | 6 |  |  | 0.0829 | 1442 | uMbRAE | 0.1176 | 0.1287 | 0.1232 | 0.1457 | 0.5833 | 0.2197 |
| 70%-30% | 6 |  |  | 0.1068 | 1418 | MAE | 0.0118 | 0.0235 | 0.0235 | 0.0118 | 0.0118 | 0.0165 |
| 70%-30% | 6 |  |  | 0.1068 | 1418 | RMSE | 0.1085 | 0.1534 | 0.1534 | 0.1085 | 0.1085 | 0.1264 |
| 70%-30% | 6 |  |  | 0.1068 | 1418 | uMbRAE | 0.1184 | 0.1258 | 0.1258 | 0.1486 | 0.5888 | 0.2215 |
| **60%-40%** | **7** | **6** |  | **0.0488** | **1635** | **MAE** | **0.0088** | **0.0000** | **0.0000** | **0.0088** | **0.0088** | **0.0053** |
| **60%-40%** | **7** | **6** |  | **0.0488** | **1635** | **RMSE** | **0.0937** | **0.0000** | **0.0000** | **0.0937** | **0.0937** | **0.0562** |
| **60%-40%** | **7** | **6** |  | **0.0488** | **1635** | **uMbRAE** | **0.1287** | **0.1287** | **0.1014** | **0.1457** | **0.6056** | **0.2220** |
| 60%-40% | 7 | 6 | 5 | 0.1244 | 1829 | MAE | 0.0263 | 0.0351 | 0.0088 | 0.0088 | 0.0088 | 0.0175 |
| 60%-40% | 7 | 6 | 5 | 0.1244 | 1829 | RMSE | 0.1622 | 0.1873 | 0.0937 | 0.0937 | 0.0937 | 0.1261 |
| 60%-40% | 7 | 6 | 5 | 0.1244 | 1829 | uMbRAE | 0.1287 | 0.1400 | 0.1014 | 0.1343 | 0.6056 | 0.2220 |
| 60%-40% | 7 | 6 | 4 | 0.1104 | 1827 | MAE | 0.0175 | 0.0263 | 0.0175 | 0.0088 | 0.0351 | 0.0211 |
| 60%-40% | 7 | 6 | 4 | 0.1104 | 1827 | RMSE | 0.1325 | 0.1622 | 0.1325 | 0.0937 | 0.1873 | 0.1416 |
| 60%-40% | 7 | 6 | 4 | 0.1104 | 1827 | uMbRAE | 0.1176 | 0.1287 | 0.1232 | 0.1457 | 0.6056 | 0.2242 |
| 70%-30% | 7 | 6 | 6 | 0.0942 | 2399 | MAE | 0.0353 | 0.0235 | 0.0000 | 0.0118 | 0.0000 | 0.0141 |
| 70%-30% | 7 | 6 | 6 | 0.0942 | 2399 | RMSE | 0.1879 | 0.1534 | 0.0000 | 0.1085 | 0.0000 | 0.0899 |
| 70%-30% | 7 | 6 | 6 | 0.0942 | 2399 | uMbRAE | 0.1333 | 0.1258 | 0.0968 | 0.1486 | 0.5888 | 0.2187 |
| 80%-20% | 7 | 6 | 3 | 0.0788 | 3763 | MAE | 0.0000 | 0.0000 | 0.0179 | 0.0179 | 0.0357 | 0.0143 |
| 80%-20% | 7 | 6 | 3 | 0.0788 | 3763 | RMSE | 0.0000 | 0.0000 | 0.1336 | 0.1336 | 0.1890 | 0.0912 |
| 80%-20% | 7 | 6 | 3 | 0.0788 | 3763 | uMbRAE | 0.1089 | 0.1200 | 0.1200 | 0.1546 | 0.6471 | 0.2301 |
| 80%-20% | 7 | 6 | 4 | 0.0583 | 2953 | MAE | 0.0179 | 0.0000 | 0.0000 | 0.0179 | 0.0179 | 0.0107 |
| 80%-20% | 7 | 6 | 4 | 0.0583 | 2953 | RMSE | 0.1336 | 0.0000 | 0.0000 | 0.1336 | 0.1336 | 0.0802 |
| 80%-20% | 7 | 6 | 4 | 0.0583 | 2953 | uMbRAE | 0.1313 | 0.1200 | 0.0980 | 0.1546 | 0.6000 | 0.2208 |

Coding in R

1. library(readxl)
3. #Import data
5. Data <- read\_excel("D:/../Set\_1.xlsx")
7. Data[is.na(Data)] <- 0
9. # Normalize the data
10. normalize <- function (x){
11. return((x - min(x)) / (max(x)-min(x)))
12. }
13. Data\_Nom = as.data.frame(lapply(Data, normalize))

16. # Split the dataset (Train & Test)
17. library(caTools)
18. set.seed(123)
19. split = sample.split(Data\_Nom, SplitRatio = 0.6)
20. TrainSet = subset(Data\_Nom, split == TRUE)
21. TestSet = subset(Data\_Nom, split == FALSE)
22. #Create Neural Network
23. library(neuralnet)
24. set.seed(123)
25. nn = neuralnet(O1 + O2 + O3 + O4 + O5 + O6 ~ Vw + G + Ta + T1 + T2, data = TrainSet, hidden = c(4,3), linear.output = FALSE, threshold = 0.01)
26. plot(nn)
28. # Test the resulting output
29. temp\_test = subset(TestSet, select = c("Vw", "G", "Ta", "T1","T2"))
30. nn\_results = compute(nn, temp\_test)
31. prediction = nn\_results$net.result
33. # Evaluation of the Accuracy
34. library(FuzzyR)
35. A = data.frame(TestSet$O1,TestSet$O2, TestSet$O3,TestSet$O4, TestSet$O5, TestSet$O6)
36. B = data.matrix(A)
37. prediction = nn\_results$net.result
38. prediction1 = round(prediction, digits = 0)
39. actual1 = round(B, digits = 0)
40. fuzzyr.accuracy(prediction1, actual1)