R_Deep_Learning_Bank_Notes

Ulises Jose Bustamante Mora

2023-11-27

Importing the dataset from a csv file

```
df = read.csv("bank_note_data.csv")
head(df)
```

```
##   Image.Var Image.Skew Image.Curt  Entropy Class
## 1   3.62160     8.6661    -2.8073 -0.44699     0
## 2   4.54590     8.1674    -2.4586 -1.46210     0
## 3   3.86600    -2.6383     1.9242  0.10645     0
## 4   3.45660     9.5228    -4.0112 -3.59440     0
## 5   0.32924    -4.4552     4.5718 -0.98880     0
## 6   4.36840     9.6718    -3.9606 -3.16250     0
```

Splitting the dataset into training and testing

```
library(caTools)
split = sample.split(df$Class, SplitRatio = 0.7)
train_df = subset(df, split ==T)
test_df = subset(df, split ==F)
```

Dimensions of each test

```
dim(train_df)
```

```
## [1] 960   5
```

```
dim(test_df)
```

```
## [1] 412   5
```

```
names(df)
```

```
## [1] "Image.Var"  "Image.Skew" "Image.Curt" "Entropy"    "Class"
```

Training our model

```
library(neuralnet)

nn = neuralnet(Class~Image.Var+Image.Skew+Image.Curt+Entropy,
               data = train_df, hidden = c(8,4,2), linear.output = F)

plot(nn)
```

Making predictions

```
nn_pre = compute(nn, test_df[1:4])

head(nn_pre$net.result)
```

```
##           [,1]
## 2  0.004142058
## 3  0.004142065
## 4  0.004142061
## 15 0.004142064
## 17 0.004142061
## 20 0.004142058
```

Rounding the results for the classification model

```
nn_pre_rounded = sapply(nn_pre$net.result,round)
head(nn_pre_rounded)
```

```
## [1] 0 0 0 0 0 0
```

Getting the confusion matrix

```
table(nn_pre_rounded, test_df$Class)
```

```
##
## nn_pre_rounded   0   1
##              0 229   0
##              1   0 183
```

We got as a results a 100%, so let's do another ML model to see if something were wrong

Training a Random Forest ML model

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.
```

Predictons

```
rf = randomForest(Class~., data=train_df)
rf_pre = predict(rf, test_df)
```

Confusion Matrix

```
table(rf_pre, test_df$Class)

##
## rf_pre   0   1
##      0 227   1
##      1   2 182
```

The Random Forest ML predicted pretty good, so we can confirm that the dataset have values very predictable, for that reason the accurate rate is quite high.