

Neural Net

Henri Makika

May 13, 2019

Datasets and graphs for the article Neural Networks - A Worked Example

```
options(scipen = 20)
```

Packages

```
suppressMessages(library(data.table))
suppressMessages(library(ggplot2))
suppressMessages(library(stringr))
```

Helper methods (Sigmoid fonctions)

```
sigmoid <- function(x){
  1 / (1 + exp(-x))
}

softmax <- function(m, offset_trick = TRUE){
  # softmax

  if(offset_trick){
    rowmaxs <- apply(m, 1, max)
    result <- exp(m - rowmaxs)/rowSums(exp(m - rowmaxs))
  } else{
    result <- exp(m)/rowSums(exp(m))
  }

  return(result)
}
```

DataSet (Read, train & test data)

```
train = fread("https://raw.githubusercontent.com/ben519/MLPB/master/Problems/Classify%20Images%20of%20S")
test <- fread("https://raw.githubusercontent.com/ben519/MLPB/master/Problems/Classify%20Images%20of%20S")
```

Insert label for plots

```

train[, Label := ifelse(IsStairs, "Stairs", "Not Stairs")]
train[, Label := factor(Label, levels=c("Not Stairs", "Stairs"))]
test[, Label := ifelse(IsStairs, "Stairs", "Not Stairs")]
test[, Label := factor(Label, levels=c("Not Stairs", "Stairs"))]

```

Remodeler de large à grand

```

trainTall <- melt(
  train,
  id.vars = c("ImageId", "IsStairs", "Label"),
  measure.vars = c("R1C1", "R1C2", "R2C1", "R2C2"),
  value.name = "Intensity",
  variable.name = "Pixel"
)

testTall <- melt(
  test,
  id.vars = c("ImageId", "IsStairs", "Label"),
  measure.vars = c("R1C1", "R1C2", "R2C1", "R2C2"),
  value.name = "Intensity",
  variable.name = "Pixel"
)

trainTall[, `:=`(Row = as.integer(str_extract(Pixel, "(?<=R)\\d")), Col = as.integer(str_extract(Pixel, "(?<=C)\\d")))]
testTall[, `:=`(Row = as.integer(str_extract(Pixel, "(?<=R)\\d")), Col = as.integer(str_extract(Pixel, "(?<=C)\\d")))]

```

Figures (fig1 = stairs vs not stairs)

```

figldata <- trainTall[ImageId %in% train$ImageId[1:12]]

fig1 <- ggplot(figldata, aes(x = Col, y = 2-Row, fill = Intensity)) + geom_tile(color = "white") + scale_fill_viridis()
print(fig1)

```

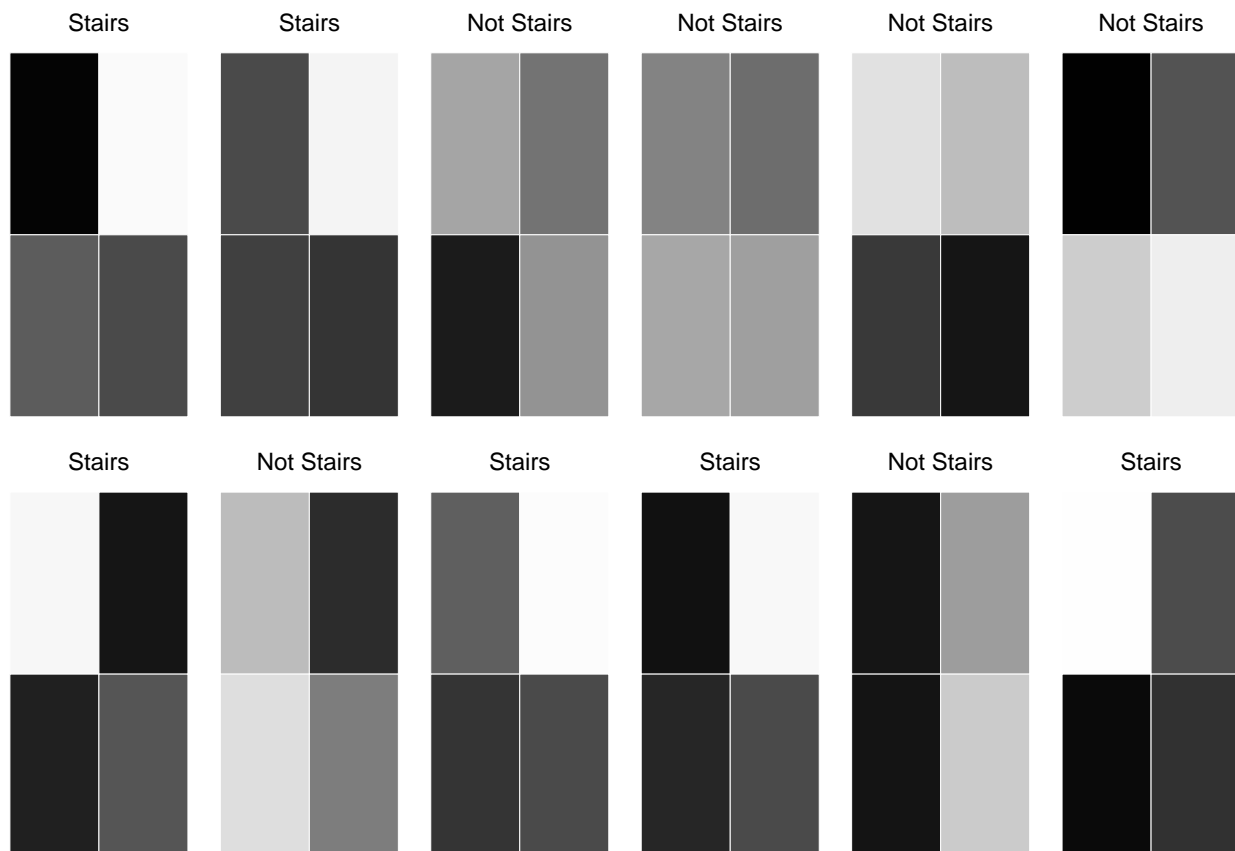
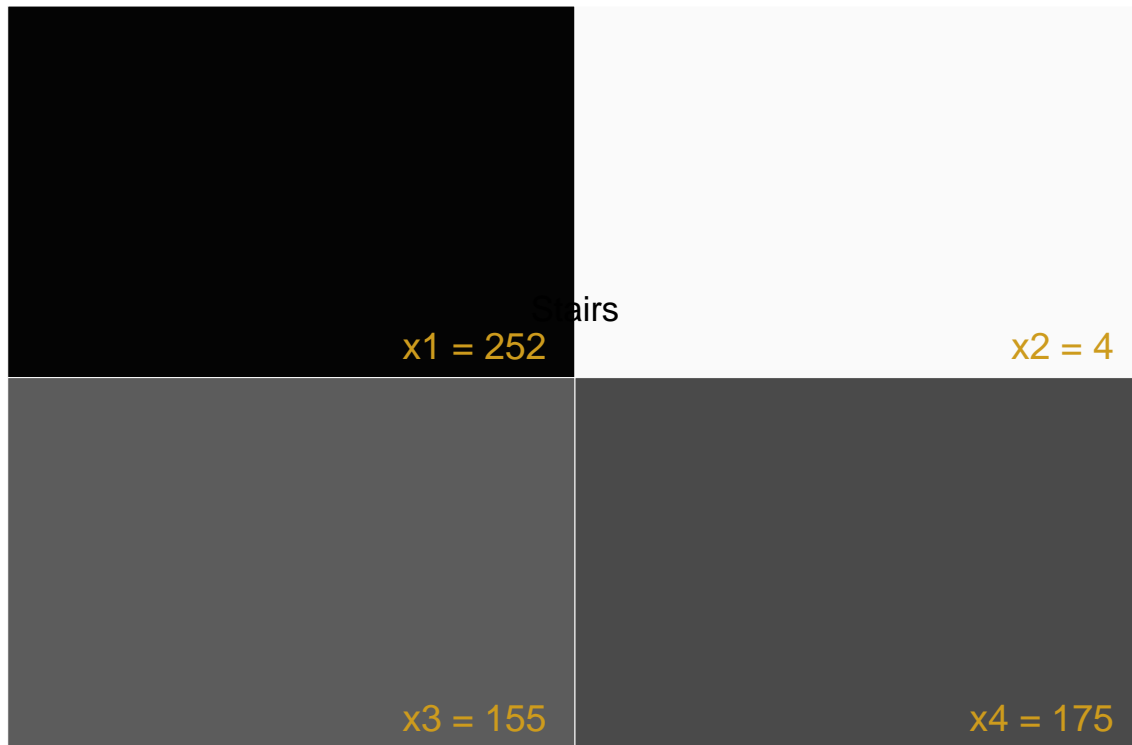


fig2: image with pixels labeled x1, x2, x3, x4

```
image1 <- trainTall[ImageId == 1]
image1[Row == 1 & Col == 1, PixelLabel := paste0("x1 = ", Intensity)]
image1[Row == 1 & Col == 2, PixelLabel := paste0("x2 = ", Intensity)]
image1[Row == 2 & Col == 1, PixelLabel := paste0("x3 = ", Intensity)]
image1[Row == 2 & Col == 2, PixelLabel := paste0("x4 = ", Intensity)]

fig2lables <- image1
fig2lables[, `:=`(x = Col+0.45, y = 1.55 - Row)]

fig2 <- ggplot(image1, aes(x = Col, y = 2-Row, fill = Intensity)) + geom_tile(color = "white") + geom_t
  labs(title = image1$Label[1]) + theme(plot.title = element_text(hjust = 0.5, vjust = -50))
print(fig2)
```



NNet from scratch

Build a nnet with an input layer, hidden layer, and output layer input layer: 1st node = 1 (for bias), nodes 2-4 correspond to features R1C1, R1C2, R2C1, R2C2 hidden layer: 3 nodes. 1st node = 1 (for bias), nodes 2-3 correspond to incoming signals output layer: 2 nodes Optimize categorical cross entropy error Apply sigmoid activation to the hidden layer, softmax to the output layer