CS422 - HW6

Code

Jane Downer

Hide

```
rm(list=ls())
library(ggplot2)
library(gganimate)
library(geometry)
```

Set working directory as needed

Hide

```
setwd("/Users/user/Downloads")
```

```
The working directory was changed to /Users/user/Downloads inside a noteb
ook chunk. The working directory will be reset when the chunk is finished
running. Use the knitr root.dir option in the setup chunk to change the w
orking directory for notebook chunks.
```

Hide

```
points <- read.csv("perceptron.csv", sep=",", header=T)
```

The Perceptron function

Hide

```
perceptron <- function(points, lamda, gamma) {
# --- Your code goes below ---
  x <- points[,2:4]
  x <- as.matrix(x)
  y <- points[,1]

  avg_list <- c()
  k <- 0
  n <- nrow(x)
  epoch <- 0
  sum_error <- 0
  weight_list <- c()
  weights <- round(runif(3, min=-1, max=1),3)
  bias <- c(weights[1])
  w1 <- c(weights[2])
  w2 <- c(weights[3])

  repeat{
    sum_error <- 0
    for(i in 1:n) {
      x_vec <- c(as.numeric(x[i,1]), as.numeric(x[i,2]), as.numeric(x[i,3
]))
      y_hat <- my_sign(weights,x_vec)
      error <- (y[i] - y_hat)
      sum_error <- sum_error + abs(error)

      for(j in 1:3){
        weights[j] <- weights[j] + (lamda*error*x_vec[j])
      }
      k <- k+1
    }
    epoch <- epoch + 1
    bias[[epoch+1]] <- weights[1]
    w1[[epoch+1]] <- weights[2]
    w2[[epoch+1]] <- weights[3]
    avg_list[epoch] <- avg <- as.numeric(format(round(sum_error/n, digits
= 3), nsmall = 2))
    if (avg <= gamma) {
      break
    }
  }
  dx <<- data.frame("bias" = bias, "w1" = w1, "w2" = w2)
  ret = list("weights" = weights, "epochs" = epoch, "error" = avg_list)
  return(ret)
}
```

The sign function

Hide

```
my_sign <- function(x, weights) {
# --- Your code goes below ---
  if (dot(x, weights) < 0) {
    return(-1)
  } else if (dot (x, weights) > 0) {
    return(1)
  } else {
    return(0)
  }
}
```

MAIN ENTRY POINT

Run perceptron function on input and return weights, epoch, and error

Hide

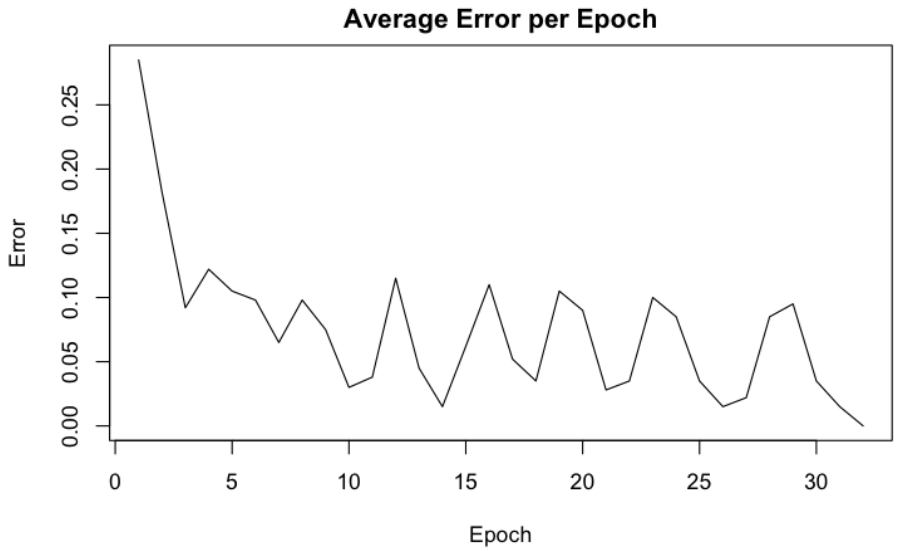
```
val = perceptron(points, 1, 0.001)
val
```

```
$weights
[1] -32.988  32.936  32.993

$epochs
[1] 32

$error
 [1] 0.285 0.182 0.092 0.122 0.105 0.098 0.065 0.098 0.075 0.030 0.038 0.
115 0.045 0.015
[15] 0.062 0.110 0.052 0.035 0.105 0.090 0.028 0.035 0.100 0.085 0.035 0.
015 0.022 0.085
[29] 0.095 0.035 0.015 0.000
```

Display plot of average error over epochs

Hide

```
plot(1:val[[2]], val[[3]], type = "l", xlab = "Epoch", ylab = "Error", ma
in = "Average Error per Epoch")
```



Prepare data for animation and run animation function

Hide

```
df1 <- subset(points, points$label == 1)
df2 <- subset(points, points$label == -1)
names(df2)[names(df2) == "x1"] <- "x1_"
names(df2)[names(df2) == "x2"] <- "x2_"

for (i in 1:nrow(dx)) {
  dx$slope__[i] <- -dx[i,2]/dx[i,3]
  dx$intercept__[i] <- -dx[i,1]/dx[i,3]
  dx$count[i] = i
}

p_base <- ggplot(data=df1, aes(x1, x2, colour = "red")) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  geom_point(data=df2, aes(x1_, x2_, colour = "green"),
             alpha = 0.7, show.legend = FALSE) +
  geom_abline(aes(slope = slope__, intercept = intercept__), data = dx)

p_base + transition_states(count, transition_length = 3, state_length =
0.001, wrap = F) +
  labs(title = 'Epoch {closest_state}') +
  ease_aes("linear")
```

```
Frame 1 (1%)
Frame 2 (2%)
Frame 3 (3%)
Frame 4 (4%)
Frame 5 (5%)
Frame 6 (6%)
Frame 7 (7%)
Frame 8 (8%)
Frame 9 (9%)
Frame 10 (10%)
Frame 11 (11%)
Frame 12 (12%)
Frame 13 (13%)
Frame 14 (14%)
Frame 15 (15%)
Frame 16 (16%)
Frame 17 (17%)
Frame 18 (18%)
Frame 19 (19%)
Frame 20 (20%)
Frame 21 (21%)
Frame 22 (22%)
Frame 23 (23%)
Frame 24 (24%)
Frame 25 (25%)
Frame 26 (26%)
Frame 27 (27%)
Frame 28 (28%)
Frame 29 (29%)
Frame 30 (30%)
Frame 31 (31%)
Frame 32 (32%)
Frame 33 (33%)
Frame 34 (34%)
Frame 35 (35%)
Frame 36 (36%)
Frame 37 (37%)
Frame 38 (38%)
Frame 39 (39%)
Frame 40 (40%)
Frame 41 (41%)
Frame 42 (42%)
Frame 43 (43%)
Frame 44 (44%)
Frame 45 (45%)
Frame 46 (46%)
Frame 47 (47%)
Frame 48 (48%)
Frame 49 (49%)
Frame 50 (50%)
Frame 51 (51%)
Frame 52 (52%)
Frame 53 (53%)
Frame 54 (54%)
Frame 55 (55%)
Frame 56 (56%)
Frame 57 (57%)
Frame 58 (58%)
Frame 59 (59%)
Frame 60 (60%)
Frame 61 (61%)
Frame 62 (62%)
Frame 63 (63%)
Frame 64 (64%)
Frame 65 (65%)
Frame 66 (66%)
Frame 67 (67%)
Frame 68 (68%)
Frame 69 (69%)
Frame 70 (70%)
```

```
Frame 71 (71%)
Frame 72 (72%)
Frame 73 (73%)
Frame 74 (74%)
Frame 75 (75%)
Frame 76 (76%)
Frame 77 (77%)
Frame 78 (78%)
Frame 79 (79%)
Frame 80 (80%)
Frame 81 (81%)
Frame 82 (82%)
Frame 83 (83%)
Frame 84 (84%)
Frame 85 (85%)
Frame 86 (86%)
Frame 87 (87%)
Frame 88 (88%)
Frame 89 (89%)
Frame 90 (90%)
Frame 91 (91%)
Frame 92 (92%)
Frame 93 (93%)
Frame 94 (94%)
Frame 95 (95%)
Frame 96 (96%)
Frame 97 (97%)
Frame 98 (98%)
Frame 99 (99%)
Frame 100 (100%)
Finalizing encoding... done!
```

The final hyper-plane (for reference)

Hide

```
d_final <- dx[nrow(dx),]
p_final <- ggplot(data=df1, aes(x1, x2, colour = "red")) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  geom_point(data=df2, aes(x1_, x2_, colour = "green"),
             alpha = 0.7, show.legend = FALSE) +
  geom_abline(aes(slope = slope__, intercept = intercept__), data = d_fin
al) +
  labs(title = "Final Hyperplane")
p_final
```