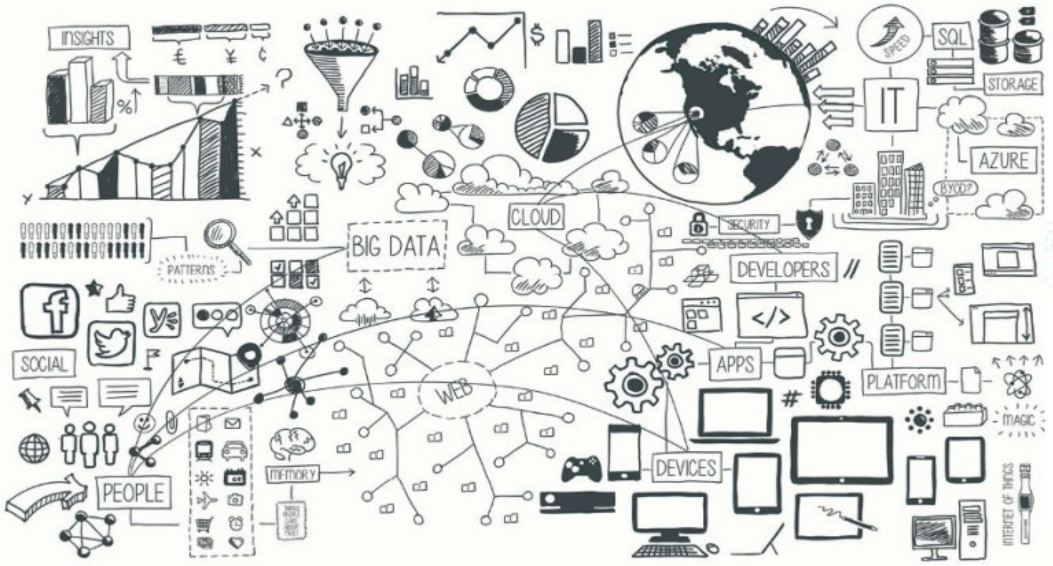
Data Mining (Minería de Datos)

Práctica: Extreme Learning Machine (ELM)



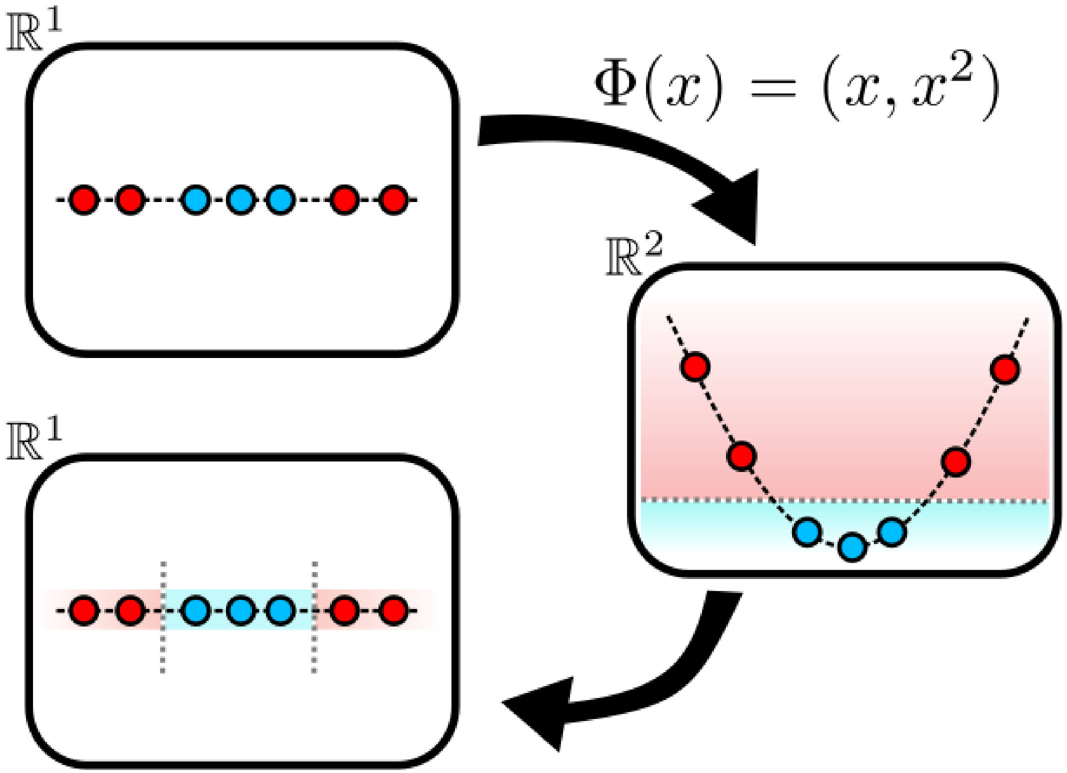
Jorge Baño

Grupo de Meteorología

Univ. de Cantabria – CSIC
MACC / IFCA

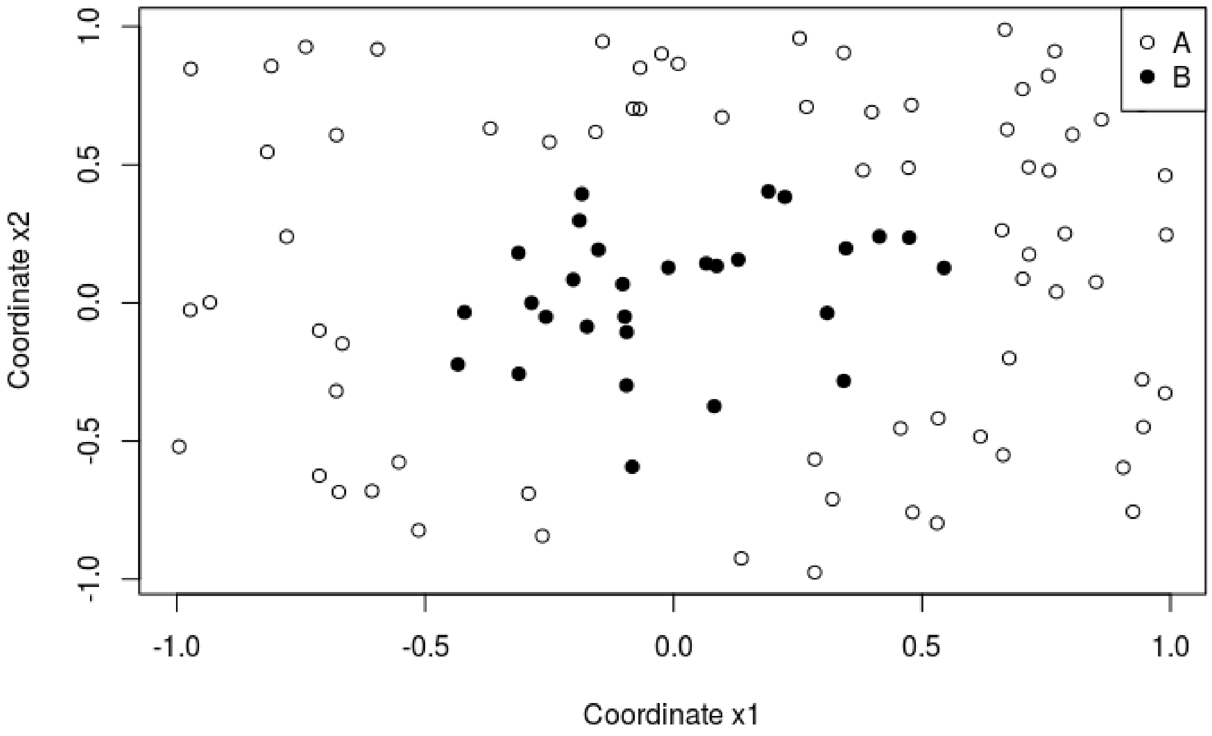
Why this works?

The problem is now linearly separable in a nonlinear subspace of the original one.



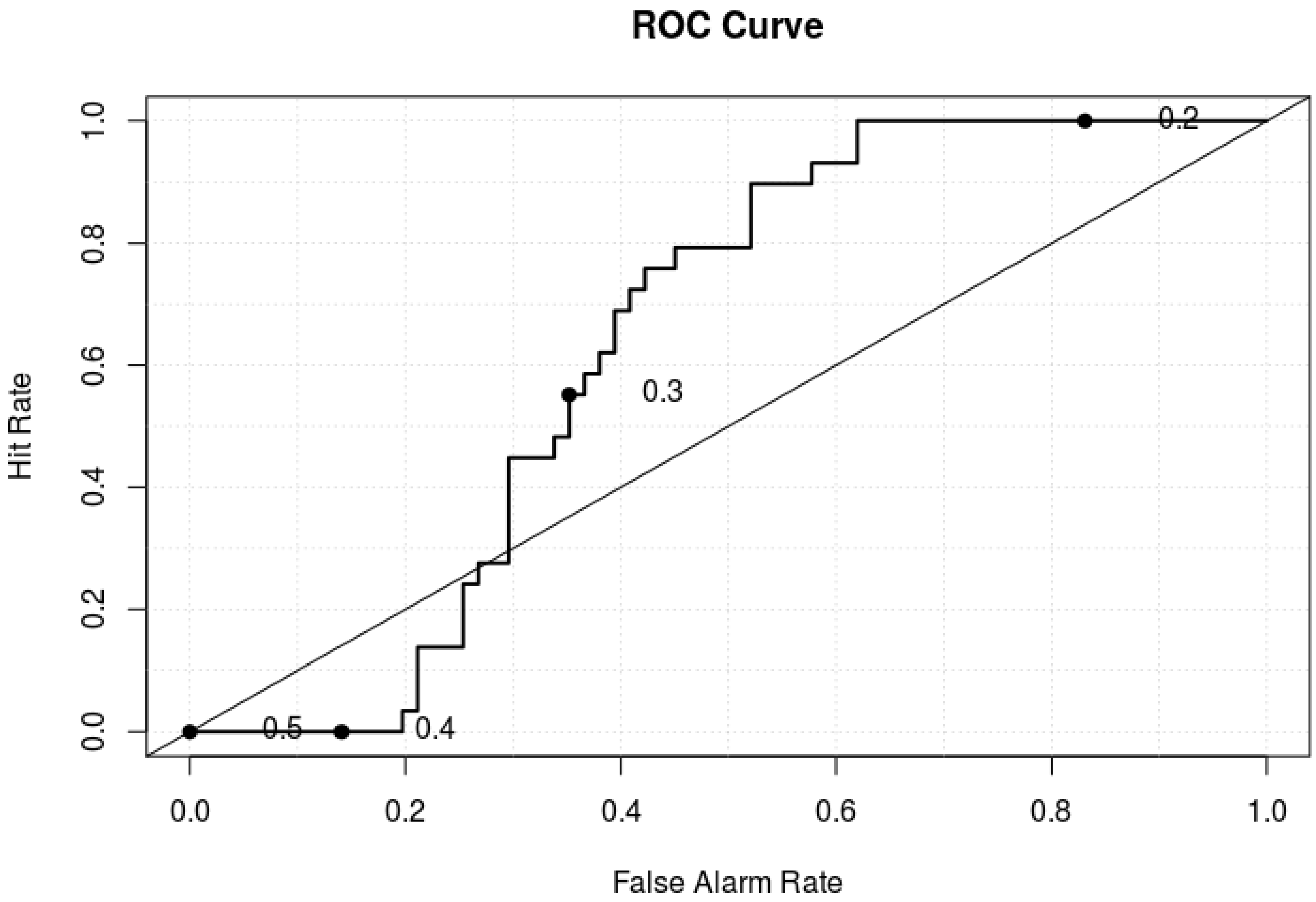
Circle problem with Extreme Learning Machines – Visualizing the dataset

```
### Cargamos las librerías
library(elmNN) # para elm
library(verification) # para roc.area
library(magrittr) # para %>%
################ PRIMERA PARTE ################
### Cargamos los datos
df <-read.table("/home/jorge/Escritorio/DataMining/MasterDataScience/circle2.txt")
df <- data.frame(cbind(y,x))
colnames(df) <- c("x1","x2","y")
### Visualizamos los datos
ind.A <- which(df$y == 0)
ind.B <- which(df$y == 1)
plot(df$x1[ind.A],df$x2[ind.A],
pch = 21, xlab = "Coordinate x1",
ylab = "Coordinate x2")
points(df$x1[ind.B],df$x2[ind.B],
pch = 19)
legend("topright",
legend = c("A", "B"),
      pch = c(21,19))
```



Circle problem with Extreme Learning Machines – Logistic Regression

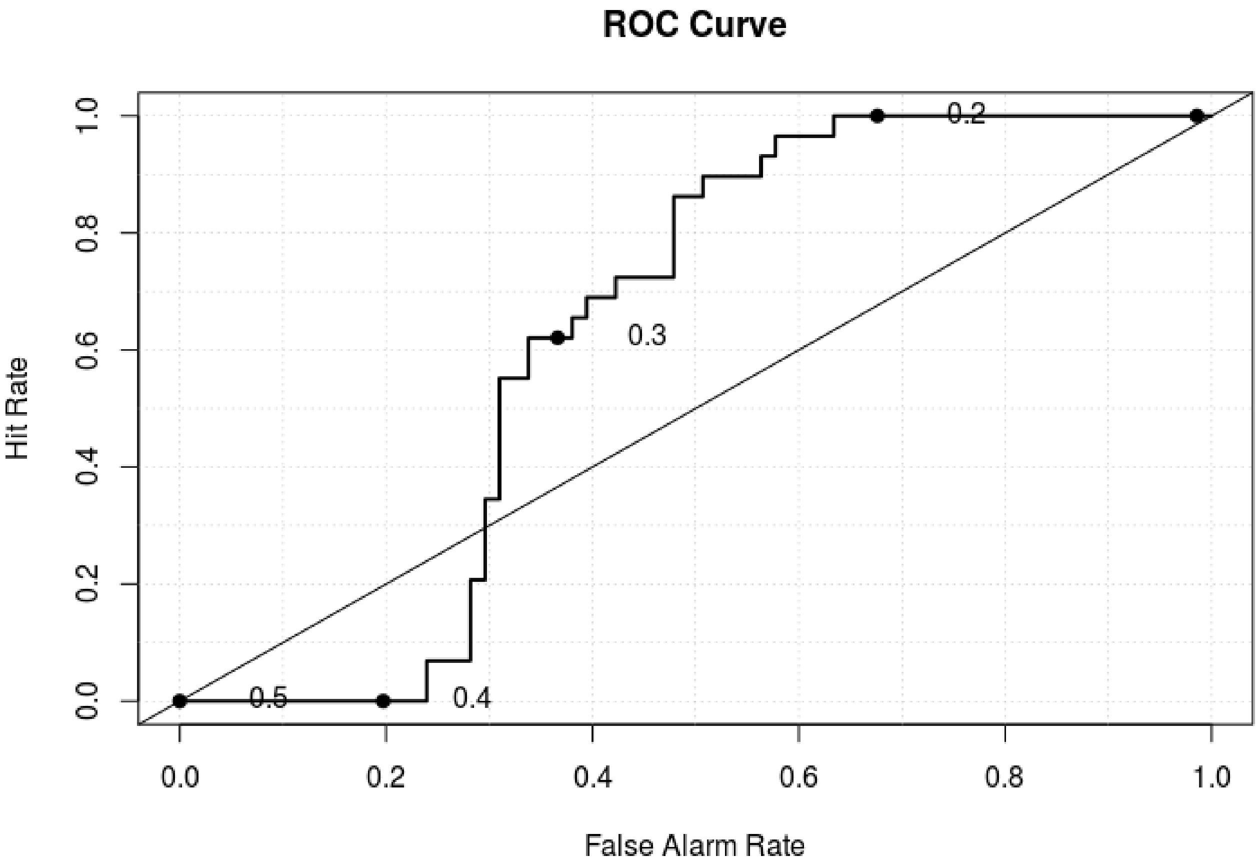
```
model <- glm(formula = y ~ ., data = df, family = binomial(link = "logit"))
pred <- predict(model, df, type = "response")
auc.sig <- roc.area(df$y,pred)$A
roc.plot(df$y,pred)
```



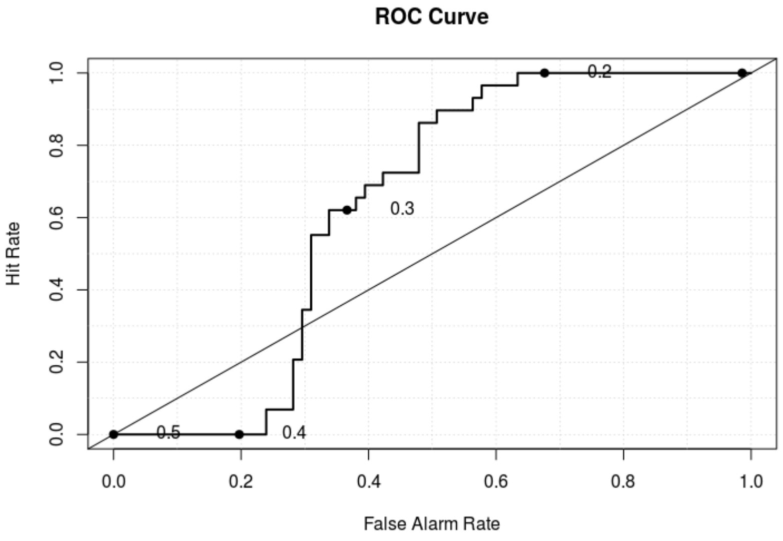
Circle problem with Extreme Learning Machines – Extreme Learning Machine

```
model <- elmtrain(x = df[,1:2], y = df[,3], nhid = 2, actfun = "sig")
pred <- predict(model, df[-3], type = "response")
roc.area(df$y,pred)$A
```

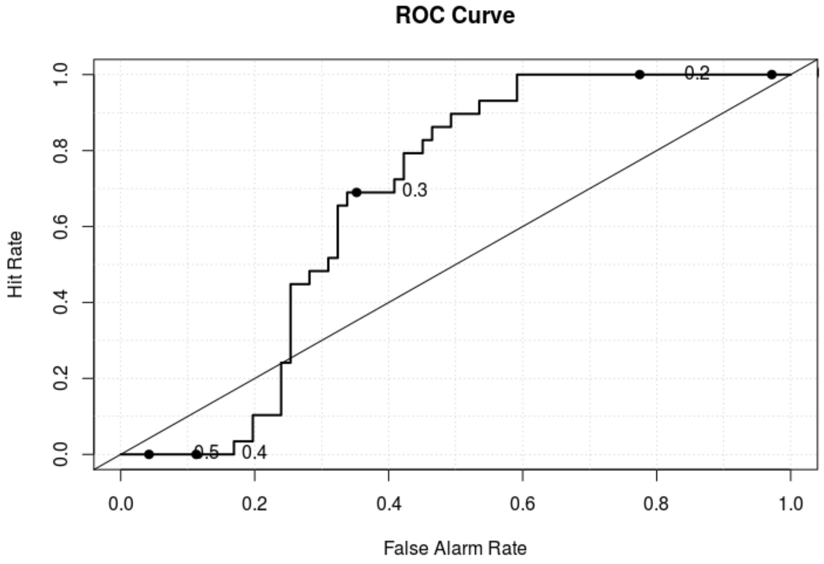
```
type of activation function.
- sig: sigmoid
- sin: sine
- radbas: radial basis
- hardlim: hard-limit
- hardlims: symmetric hard-limit
- satlins: satlins
- tansig: tan-sigmoid
- tribas: triangular basis
- poslin: positive linear
- purelin: linear
...
not used.
```



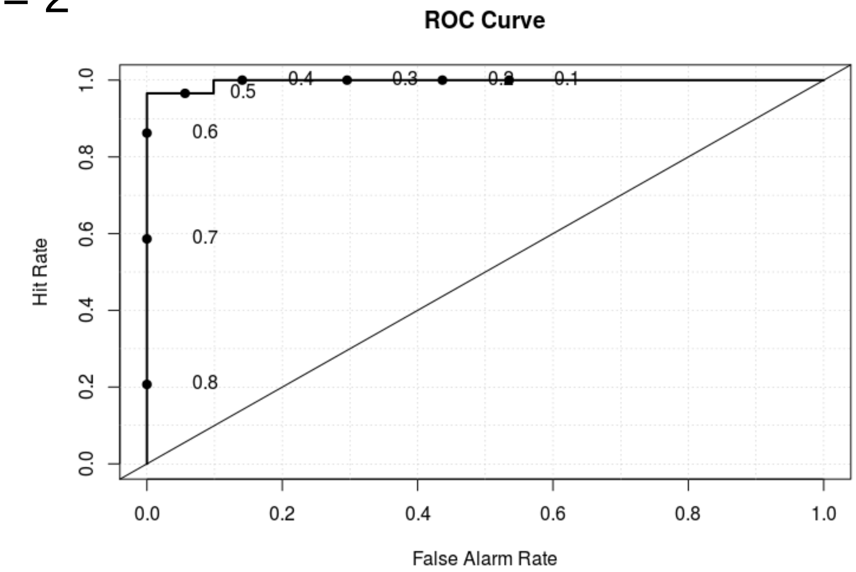
Circle problem with Extreme Learning Machines – Extreme Learning Machine



Hidden layer size = 2



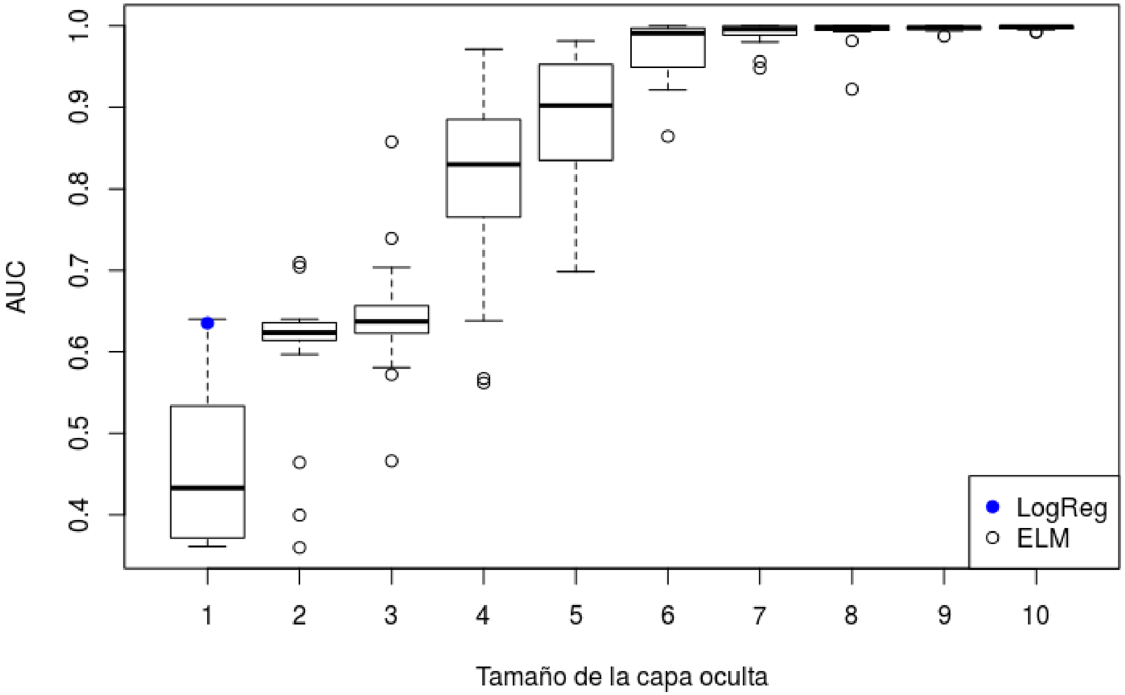
Hidden layer size = 4



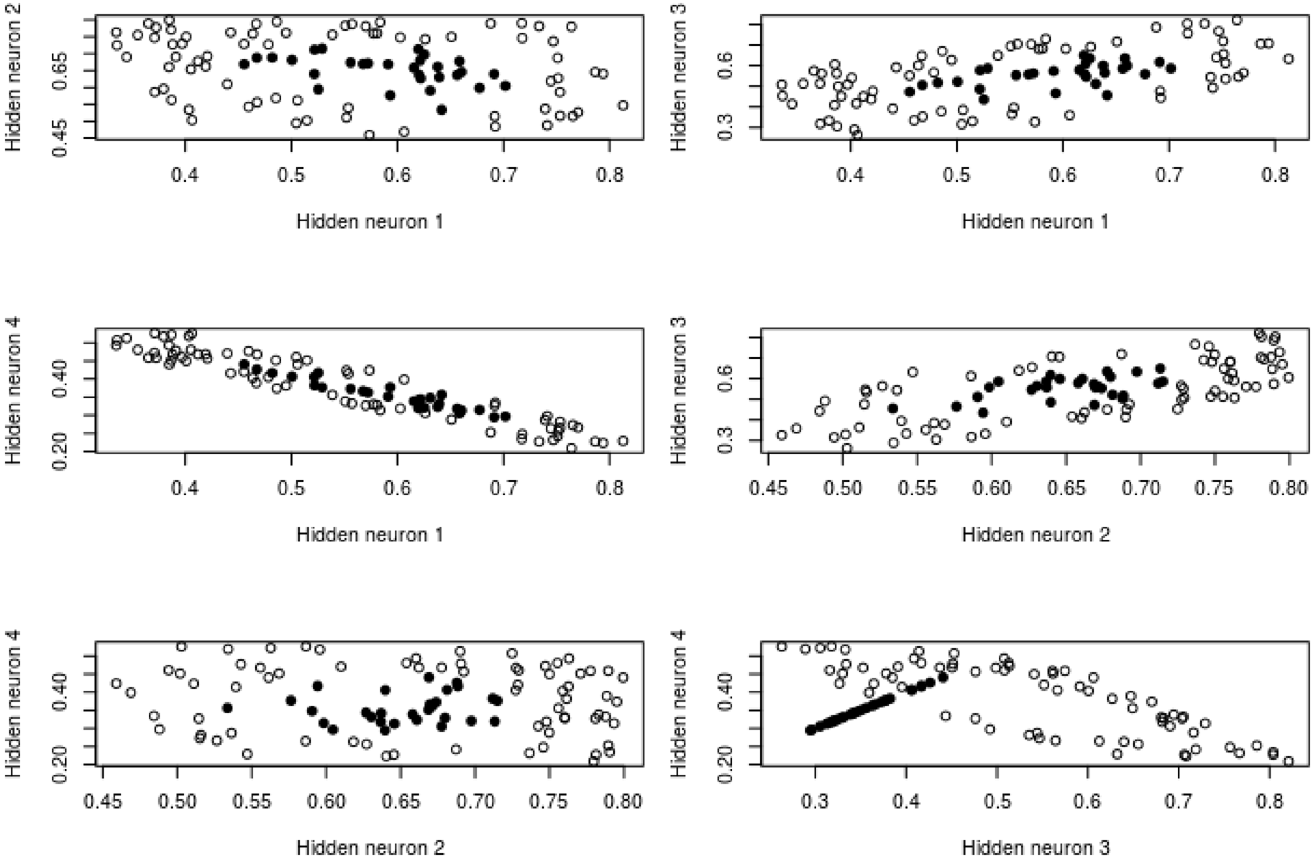
Hidden layer size = 6

Circle problem with Extreme Learning Machines – Extreme Learning Machine

```
auc.elm.rastreo <- sapply(1:20 ,FUN = function(z){ # Número de repeticiones: 20
  sapply(1:10,FUN = function(zz) { # Neuronas ocultas del 1 al 10 en pasos de 1
    model <- elmtrain(x = df[,1:2], y = df[,3], nhid = zz, actfun = "sig")
    pred <- predict(model, df[-3], type = "response")
    roc.area(df$y,pred)$A
  })
})
boxplot(t(auc.elm.rastreo), xlab = "Tamaño de la capa oculta", ylab = "AUC")
points(1,auc.sig, col = "blue", pch = 19)
legend("bottomright",
legend = c("LogReg","ELM"),
       col = c("blue","black"),
       pch = c(19,21))
```



```
model <- elmtrain(x = df[,1:2], y = df[,3], nhid = 4, actfun = "sig")
pred <- predict(model, df[-3], type = "response")
hid.neu <- 1 / (1 + exp(-t(model$inpweight %*% t(as.matrix(df[,1:2])) +
model$biashid)))
par(mfrow = c(3,2))
plot(hid.neu[ind.A,1],
hid.neu[ind.A,2], pch = 21,
 xlab = "Hidden neuron 1",
ylab = "Hidden neuron 2")
points(hid.neu[ind.B,1],
hid.neu[ind.B,2], pch = 19)
plot(hid.neu[ind.A,1],
hid.neu[ind.A,3], pch = 21,
xlab = "Hidden neuron 1",
ylab = "Hidden neuron 3")
points(hid.neu[ind.B,1],
hid.neu[ind.B,3], pch = 19)
```


Hidden neuron 2
Hidden neuron 1
Hidden neuron 3
Hidden neuron 1
Hidden neuron 4
Hidden neuron 1
Hidden neuron 3
Hidden neuron 2
Hidden neuron 4
Hidden neuron 2
Hidden neuron 4
Hidden neuron 3