Introduction to Regularization using the h2o package.

Tshiamo Phaahla

20/11/2021

Please Note The Following.

The content contained in this document is from the fourth lab session in a series of lectures prepared for a two-week introductory course in Machine Learning at the University of Cape Town, South Africa. The course is aimed at students with some background in statistical modelling, computing, and linear algebra. It is recommended that you watch the third Key-Point Lecture in this series before tackling the lab session.

Link to the the relevant lab <linkhttps://youtu.be/ue_KDc3VxLw

==========

Goal: Fit a neural network to the ptitanic data from the `rpart.plot` using `h2o`. Apply regularization and note the effect on response curves. Measure training and validation error for a given specification.

Task 1: Pre-process the data: Create training and validation frames using an 80:20 split of the data. Use only the pclass, sex, and age variables.

```
data(ptitanic)

data = na.omit(ptitanic)

set.seed(2021)

data = data.frame(survived = as.factor(data$survived), pclass = as.factor(data$pclass),
                  sex = as.factor(data$sex), age = as.numeric(data$age))

N = dim(data)[1]

#Sample from the titanic dataset without replacement
set = sample(1:N, floor(0.8*N), replace=FALSE)

data_train = as.h2o(data[set, ])
data_val = as.h2o(data[-set, ])
```

Task 2: Fit a (5,3)-network to the data using h2o.deeplearning(). Pick appropriate activation and objective functions. Set the l2 regularisation to be 0, initially. Set the learning rate to 0.001, train for 1000 epochs. Report the training and validation error.

```
# Create a data frame, split the data and cast the training and validation frames as an h2o objects:
model = h2o.deeplearning(x = 2:4, y=1, standardize =TRUE,
                         training_frame = data_train,
                         validation_frame = data_val,
                         hidden = c(5,3),
                         activation = 'Tanh',
                         distribution = "bernoulli",
                         loss = "CrossEntropy",
                         adaptive_rate = FALSE,
                         rate=0.001, epochs = 1000, l2 =0,
                         reproducible = TRUE, seed=2)

#plot(model)
report = h2o.logloss(model, train=T, valid = T)
report
```

Some Comments

- Dimensions (hidden 5 nodes first layer, 3 nodes second layer.).
- Activations function: tanh for us - only applies to hidden layers.
- Based on specification of the distribution then the algorithm will pick the appropriate activation function on the output layer. In this case it was SoftMax.
- Objective function: Need to first specify distribution, because the objective function depends on it.
- Depending on what problem you working on your response variable will determine what the things on the output layer will look like and it's interface with the objective function.
- Here we see binary classification problem so we will go with bernoulli for our distribution.
- If it were multi-class then multinomial would be the choice of our distribution.
- loss: We use cross-entropy because this is a classification problem.
- rate – learning rate: Getting into the crucks of our gradient descent. Parameters for the learning algorithm (Back-propagation with gradient descent).

A caveat with the h2o package - reproducibility

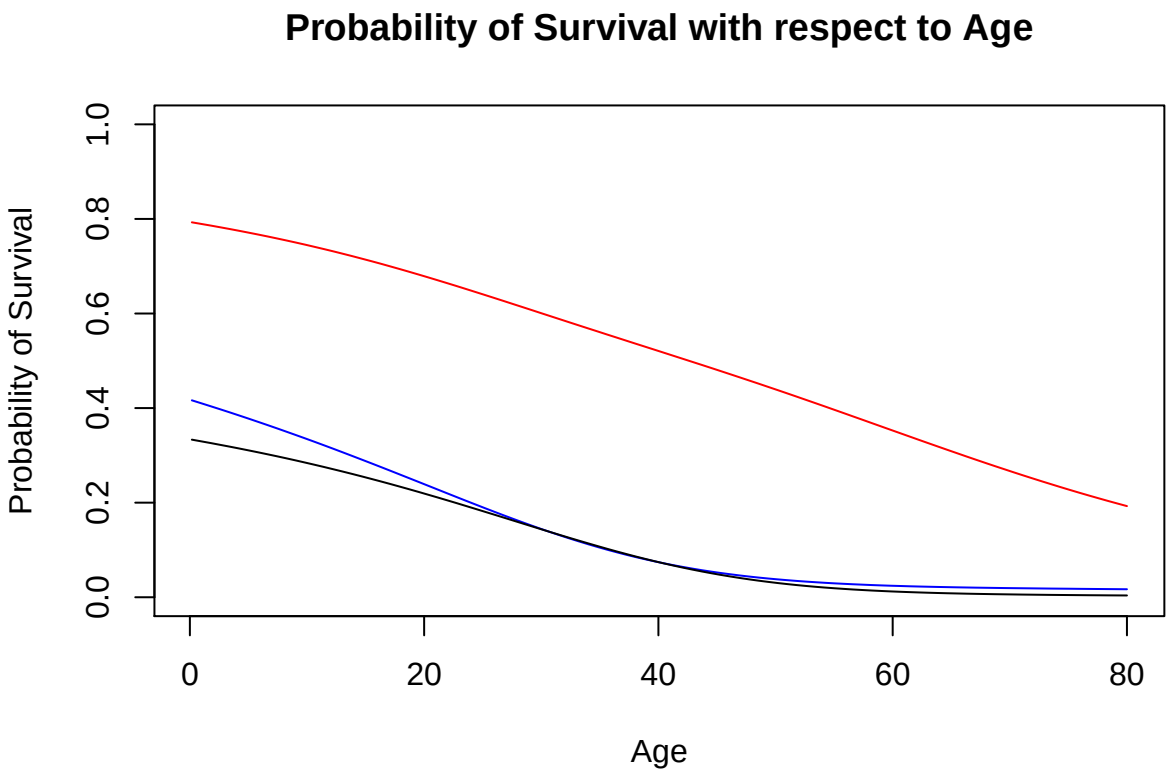
Everything runs on local cluster. The benefit of this is that we can use all the power of our machine for the problem. For extremely complex problems you can run the algorithm across a distributed network if necessary, in order to overcome the any limits to computational power which a specific researcher may have.

The problem is that the so-called "hogwild algorithm" dsitributes in such a way that the analysis over different runs of your algorithm is not actually reproducible. So basically not quite reliable.

"It is a bit strange that someone would sacrifice reproducibility for speed. Especially in the world of Science." - But there's probably a good explanation somewhere for this particular case.

Task 3: Create response curves over age for first, second and third class males. Note the effects for increasing l2.

```
M=100
ageSeq = seq(min(data$age), max(data$age), length =M)

Xnew1 = data.frame(pclass='1st', sex='male', age = ageSeq)
Xnew2 = data.frame(pclass='2nd', sex='male', age = ageSeq)
Xnew3 = data.frame(pclass='3rd', sex='male', age = ageSeq)

# create predictions based on the model:

pred1 = h2o.predict(model, as.h2o(Xnew1))
pred1 = as.data.frame(pred1)

pred2 = h2o.predict(model, as.h2o(Xnew2))
pred2 = as.data.frame(pred2)

pred3 = h2o.predict(model, as.h2o(Xnew3))
pred3 = as.data.frame(pred3)

#Plot the predictions
plot(pred1$survived ~ Xnew1$age, type='l', col="red", ylim=c(0,1), xlab = "Age",
     ylab= "Probability of Survival",
     main = "Probability of Survival with respect to Age")
lines(pred2$survived~Xnew2$age, col="blue")
lines(pred3$survived~Xnew3$age, col="black")
```



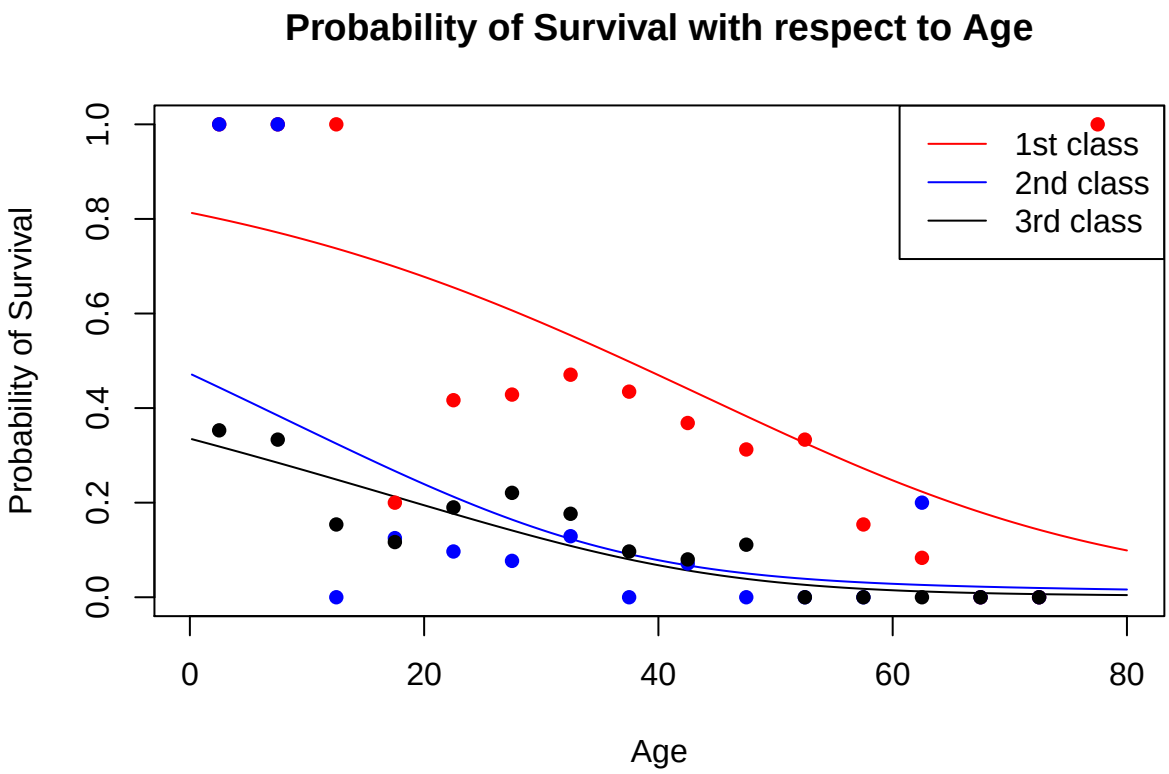
What do we see for increasing levels of l2

No plots for this included in this document. As we turn up the regularisation parameter we are further constraining our model. The more we constrain our model the less effect the actual predictors have on the response curve. The model will eventually look like the baseline survival rate because they will not distinguish between class and age etc. The appropriate level of l2 is the one which corresponds to the minimum validation error.

Task 4: Set `l2 = 0.01`. As a sanity check on the predictions, superimpose the empirical survival rates for 1st, 2nd, 3rd class males using age 'bins' of 5 years.

The only new code is creating age bins and superimposing the empirical survival rates on the response curves plot.

```
plot(pred1$survived ~ Xnew1$age, type='l', col="red", ylim=c(0,1), xlab = "Age",
     ylab= "Probability of Survival",
     main = "Probability of Survival with respect to Age")
lines(pred2$survived~Xnew2$age, col="blue")
lines(pred3$survived~Xnew3$age, col="black")
legend('topright', lty=1, legend = c('1st class', '2nd class', '3rd class'),
       col=c('red', 'blue', 'black'), bty='o')

Y = (data$survived == 'survived')

ageBins = seq(0, 80, by=5)

emp1 = rep(NA, length(ageBins)-1)
emp2 = rep(NA, length(ageBins)-1)
emp3 = rep(NA, length(ageBins)-1)

for(i in 2:length(ageBins)){
  emp1[i-1] = mean(Y[(data$sex=='male')&(data$pclass=='1st')&
                       (data$age>=ageBins[i-1])&(data$age<=ageBins[i])])
}
points(emp1~c(ageBins[-1]-2.5), col='red', pch=16)

for(i in 2:length(ageBins)){
  emp2[i-1] = mean(Y[(data$sex=='male')&(data$pclass=='2nd')&
                       (data$age>=ageBins[i-1])&(data$age<=ageBins[i])])
}
points(emp2~c(ageBins[-1]-2.5), col='blue', pch=16)

for(i in 2:length(ageBins)){
  emp3[i-1] = mean(Y[(data$sex=='male')&(data$pclass=='3rd')&
                       (data$age>=ageBins[i-1])&(data$age<=ageBins[i])])
}
points(emp3~c(ageBins[-1]-2.5), col='black', pch=16)
```



Some Interpretation

Based on the response curves, the predicted survival rates decrease for all classes quite rapidly with age. The first class passengers fair much better than second and third class passengers. With the second and third class passengers there seems to be quite little difference between their predicted survival rates for most of the age ranges although it seems that for the younger individuals, then second class passengers had a much higher likelihood/chance of survival.

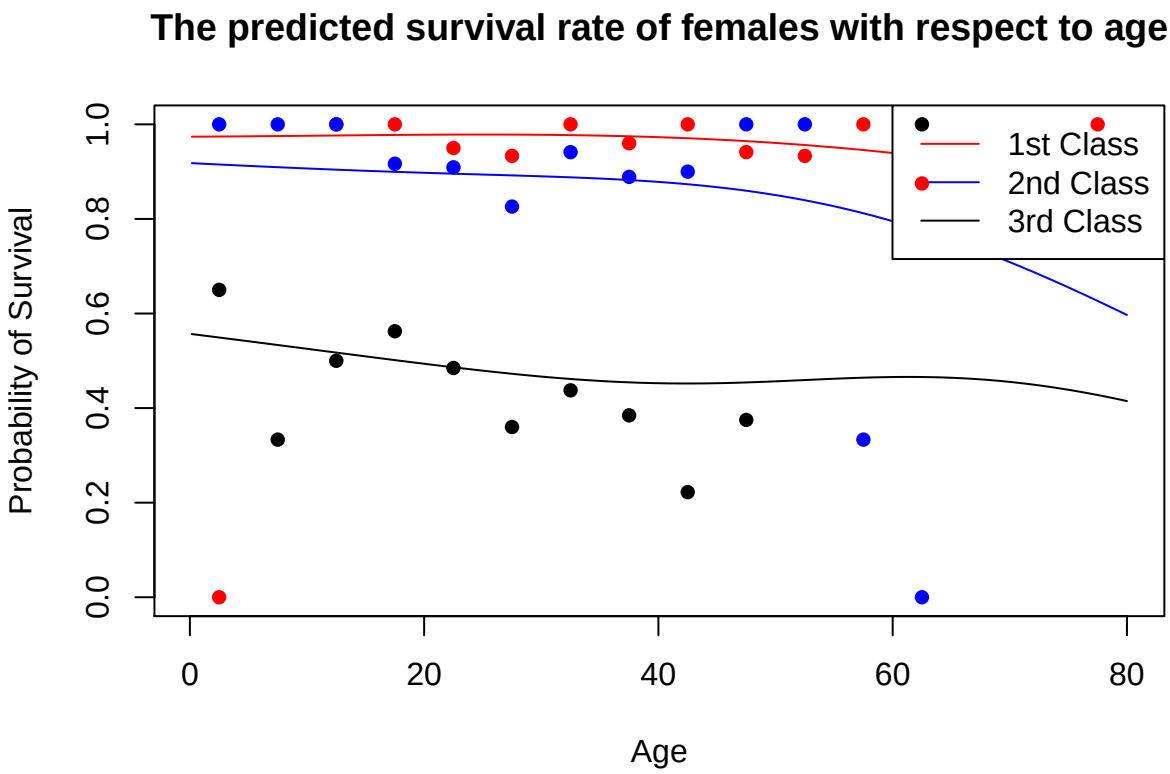
General Observation

Looking at the delta between the classes. Even though the shapes are more or less similar, decreasing as age occurs. the decreasing is less severe for the first class passengers but the delta seems to be between first class and then second and third class, where there seems to be little difference between second and third class male passengers.

Additional Exercises/Problems

Problem 1: Repeat tasks three and four but now fixing 'sex = 'female' '. Interpret the resulting response curves.

```
Xnew1f = data.frame(pclass='1st', sex='female', age = ageSeq)
Xnew2f = data.frame(pclass='2nd', sex='female', age = ageSeq)
Xnew3f = data.frame(pclass='3rd', sex='female', age = ageSeq)

pred1f = h2o.predict(model, as.h2o(Xnew1f))
pred1f = as.data.frame(pred1f)

pred2f = h2o.predict(model, as.h2o(Xnew2f))
pred2f = as.data.frame(pred2f)

pred3f = h2o.predict(model, as.h2o(Xnew3f))
pred3f = as.data.frame(pred3f)

plot(pred1f$survived~Xnew1f$age, ylim = c(0,1), col='red', type='l',
     ylab = 'Probability of Survival', xlab ='Age',
     main= "The predicted survival rate of females with respect to age")
lines(pred2f$survived~Xnew2f$age, col='blue')
lines(pred3f$survived~Xnew3f$age, col='black')
legend('topright', lty=1, legend = c('1st Class', '2nd Class', '3rd Class'),
       col=c('red', 'blue', 'black'), bty='o', bg='white')

emp1f = rep(NA, length(ageBins)-1)
emp2f = rep(NA, length(ageBins)-1)
emp3f = rep(NA, length(ageBins)-1)

for(i in 2:length(ageBins)){
  emp1f[i-1] = mean(Y[(data$pclass=='1st')&(data$sex=='female')&
                        (data$age>=ageBins[i-1])&(data$age<=ageBins[i])])
}
points(emp1f~c(ageBins[-1]-2.5), pch=16, col='red')

for(i in 2:length(ageBins)){
  emp2f[i-1] = mean(Y[(data$pclass=='2nd')&(data$sex=='female')&
                        (data$age>=ageBins[i-1])&(data$age<=ageBins[i])])
}
points(emp2f~c(ageBins[-1]-2.5), pch=16, col='blue')

for(i in 2:length(ageBins)){
  emp3f[i-1] = mean(Y[(data$pclass=='3rd')&(data$sex=='female')&
                        (data$age>=ageBins[i-1])&(data$age<=ageBins[i])])
}
points(emp3f~c(ageBins[-1]-2.5), pch=16, col='black')
```



Some Interpretation

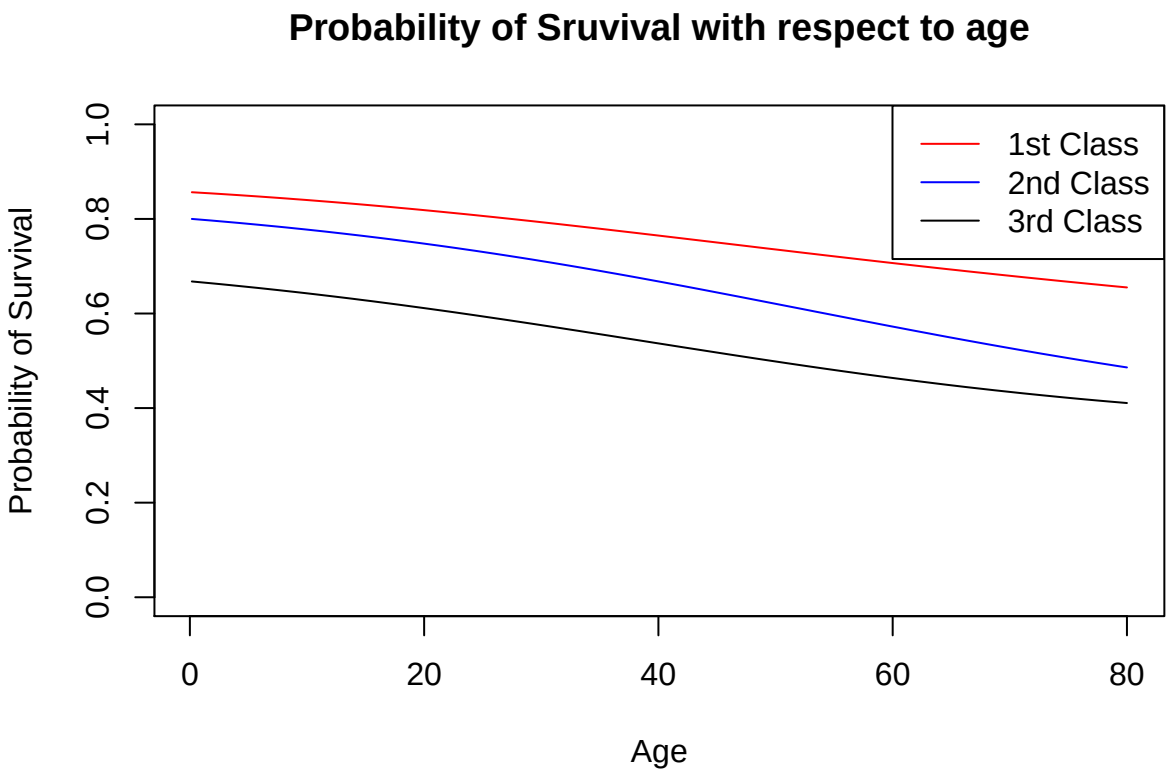
Based on the response curves the survival rates decrease with age, but not as rapidly as males. The first class passengers had the highest survival rates, followed by 2nd class passengers. The 3rd class passengers had the lowest survival rates. There appears to be little difference between the survival rates of first and 2nd class passengers. Whereas for the 3rd class passengers the surivial rates are much lower in comparison. With 3rd class passengers there survival rate response curve is not monotonic suggesting that there is no general trend of survival rates decreasing with age.

General Observation

Looking at the delta between the curves we can almost group the survival rates of 1st and 2nd class passengers and compare them with that of 3rd glass as the difference in survival rates is much higher between 3rd class and 1st and 2nd when compared to the survival rates between 1st and 2nd class passengers.

Problem 2: Repeat the analysis (all else equal) using a (10,10) - network, and 'l2=0.5'. Based on the resulting response curve, would you say that this is a 'more complex' model than a (5,3) - network ?

```
model2 = h2o.deeplearning( x = 2:4, y=1, standardize = TRUE,
                           hidden = c(10,10),
                           l2=0.5,
                           activation = 'Tanh',
                           distribution = 'Bernoulli',
                           loss = 'CrossEntropy',
                           reproducible = TRUE,
                           seed = 2,
                           rate = 0.001,
                           epochs = 1000,
                           adaptive_rate = FALSE,
                           training_frame = data_train,
                           validation_frame = data_val)

err.report = h2o.logloss(model2, train = T, valid = T)
err.report

XnewP1 = data.frame(pclass='1st', sex='female', age=ageSeq)
XnewP2 = data.frame(pclass='2nd', sex='female', age=ageSeq)
XnewP3 = data.frame(pclass='3rd', sex='female', age=ageSeq)

predP1 = h2o.predict(model2, as.h2o(XnewP1))
predP1 = as.data.frame(predP1)

predP2 = h2o.predict(model2, as.h2o(XnewP2))
predP2 = as.data.frame(predP2)

predP3 = h2o.predict(model2, as.h2o(XnewP3))
predP3 = as.data.frame(predP3)

plot(predP1$survived~XnewP1$age, type='l', col='red', ylim=c(0,1), xlab = 'Age',
     ylab = 'Probability of Survival',
     main = 'Probability of Sruvival with respect to age')
lines(predP2$survived~XnewP2$age,col='blue')
lines(predP3$survived~XnewP3$age, col = 'black')
legend('topright', lty=1, legend = c('1st Class', '2nd Class', '3rd Class'),
       col = c('red', 'blue', 'black'), bty='o')
```

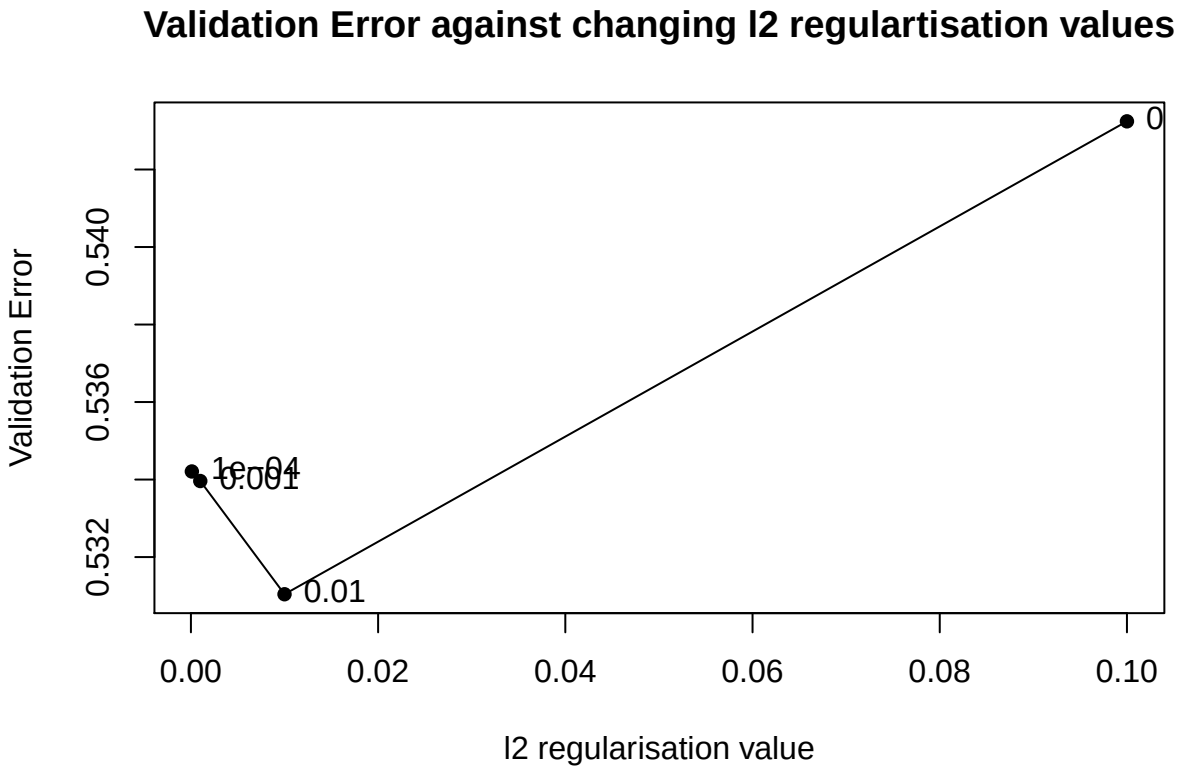


Answer

Based on the response curves the resulting model is more complex. The response curves of the (10,10) network with `l2=0.05` are much smoother than the response curves for a (5,3) network with `l2=0.1`. In addition to this there is a more general/consistent decline in the probability of survival as age increases in the (10,10) network predictions for all classes.

Problem 3: Repeat Task 2 (under a (5,3)-network), but record the validation error for 'l2 = 0.0001, 0.001, 0.01, 0.1'. Plot the resulting values as a function of these values. Which amongst these values would impose an appropriate level of regularisation?

```
l2_set = c(0.0001, 0.001, 0.01, 0.1)

val_err = rep(NA, 4)

for(i in 1:4){
  model3 = h2o.deeplearning(x = 2:4, y=1, standardize =TRUE,
                            training_frame = data_train,
                            validation_frame = data_val,
                            hidden = c(5,3),
                            activation = 'Tanh',
                            distribution = "bernoulli",
                            loss = "CrossEntropy",
                            adaptive_rate = FALSE,
                            rate=0.001, epochs = 1000, l2 = l2_set[i],
                            reproducible = TRUE, seed=2)
  v_err = h2o.logloss(model3, valid = T)
  val_err[i] = v_err
}

plot(val_err ~ l2_set, pch=16, ylim = range(val_err), xlab="l2 regularisation value",
     ylab = "Validation Error",
     main = "Validation Error against changing l2 regulartisation values")
lines(val_err ~ l2_set)
text(val_err ~ l2_set, labels=l2_set, pos=4)
```



Answer

As a rule of thumb, we pick the regularisation value which results in the lowest validation error. * In this case that is `l2=0.01`.