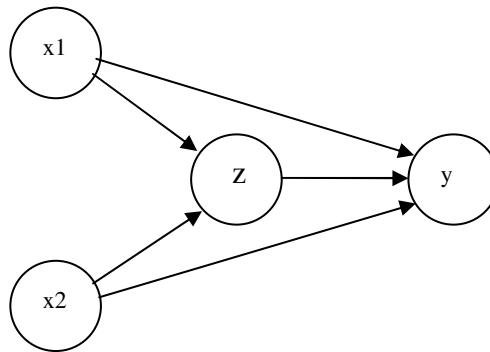


Neural Nets: Exercises/Assignment

Exercise 1. Find weights so that the neural net below reproduces the XOR function:

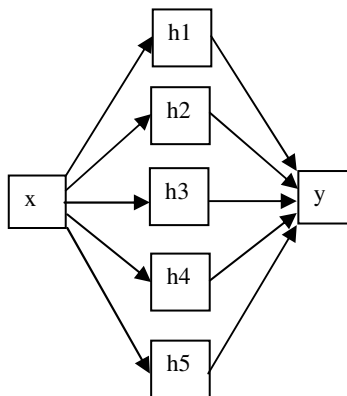
x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

Use threshold activation functions for each (non-input) node. (Bias nodes have not been shown, but can be used.)



Can you do this without node z?

Exercise 2. Find weights so that the neural net below “best” reproduces the function $y = a + bx$ for x in the range $[0, 1]$. Use threshold activation functions for the hidden layer and a linear activation function for the output node. (Bias nodes have not been shown, but can be used.)



Exercise 3: Pattern Recognition. Consider the letters A, B, C, ..., Z encoded digitally on a 7*5 grid. For example A is coded as

```
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
```

The file letters.txt gives encodings for all 26 letters. Each encoding is saved as a vector of length 35, taking each row sequentially. Use this to build a neural network which classifies binary images of size 7*5 (equivalently binary vectors of length 35) as letters of the alphabet.

The files lettersTEST.txt gives corrupted encodings for all 26 letters, for example the corrupted A is

```
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 1 1 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
```

How well does your neural net perform at classifying these corrupted images?

Use the nnet package in R to fit and test your model.

Assignment 5. Based on Kuhnert and Venables, Lab 12.

Load the rock data from the datasets package as follows

```
library(datasets)
data(rock)
?rock
```

The goal of the assignment is to model permeability as a function of area, perimeter and shape. We first transform the inputs to aid the model fitting:

```
rock.x <- data.frame(area = rock$area/10000, peri = rock$peri/10000,
                    shape = rock$shape)
rock.y <- log(rock$perm)
```

Now fit a single layer feed-forward network, with three hidden nodes, a linear output function, skip connections, and regularisation parameter $\lambda = 0.001$.

Fit the model a number of times and note how the sum of squared residuals changes each time. Why is this?

Draw your best fitted network, showing edge weights and biases. Since there is not enough data for a separate test data set, to assess the model do a plot of observed vs. predicted responses and a residual plot.

Assignment 6. Based on Kuhnert and Venables Lab 12.

Load the crabs data set from the MASS package, and have a look at it, as follows:

```
library(MASS)
data(crabs)
?crabs
pairs(crabs)
windows()
par(mfrow=c(2,2))
hist(crabs$FL)
hist(crabs$RW)
hist(crabs$CL)
hist(crabs$BD)
```

In this assignment you are to compare neural nets with logistic regression, with particular reference to the effect of model selection procedures.

Firstly fit a logistic regression model for sex with inputs sp, FL, RW, CL, CW and BD. Use all the data to fit the model, then perform 10-fold cross-validation to estimate its performance. Next using stepAIC, select a model with a reduced set of parameters and estimate its performance using cross-validation (use the same 10 subsets of the data).

Secondly fit a feed-forward neural-net with a single hidden layer of 5 nodes. Do not use any regularisation, and fit the model a number of times and take the model with best fit. Estimate the performance of this model using cross-validation (use the same subsets as before). Then refit the neural-net using a regularisation parameter value 1, and apply cross-validation as before.

Which model performs best?

The glm and nnet commands will not do cross-validation for you, but you can use the following code:

```
# generate indices for cross validation
n <- length(crabs$sex)
num.xval <- 10 # number of cross validation sets
k <- n/num.xval
xval.idx <- list()
s <- sample(1:n, n) # permutation of 1:n
for (i in 1:num.xval) {
  xval.idx[[i]] <- s[(ceiling(k*(i-1))+1):(ceiling(k*i))]
}

# apply cross validation
classerr.xval <- vector()
for (i in 1:num.xval) {
  train.set <- rep(T, n)
  train.set[xval.idx[[i]]] <- F
  test.set <- rep(F, n)
  test.set[xval.idx[[i]]] <- T
  # fit model using data=subset(crabs, train.set)
  ...
  # apply model to newdata=subset(crabs, test.set)
  crabs.fit <- ...
  # convert probs to classes; calc misclassification error
  crabs.class <- factor(crabs.glm.fit < 0.5, levels=c(T,F), labels=c('F','M'))
  classerr.xval[i] <- sum(crabs.class != crabs$sex[test.set])
    /length(xval.idx[[i]])
}
cat('xval classification error:', mean(classerr.xval),
    'with std dev', sd(classerr.xval), '\n')
```