Examination

Linköping University, Department of Computer and Information Science, Statistics

Course code and name	TDDE01 Machine Learning
Date and time	2023-01-13, 14.00-19.00
Assisting teacher	Oleg Sysoev
Allowed aids	PDF of the course book + your help file (if submitted to LISAM in due time)
Grades:	
	5=18-20 points
	4=14-17 points
	3=10-13 points
	U=0-9 points

Provide a detailed report that includes plots, conclusions and interpretations. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in the appendix.

Note: seed 12345 should be used in all codes that assumes randomness unless stated otherwise!

To start work in RStudio, type this in the Terminal application:

```
module add courses/TDDE01
rstudio
```

To submit your report:

1. Create one file (DOC, DOCX, ODT, PDF)
2. Use Exam Client to submit, and choose Assignment 1 in the drop box
3. Attach your report
4. Submit.
5. “Request Received” status implies that your report is successfully submitted.

Assignment 1 (10p)

File **Rice.csv** contains a total of 3810 rice grain's images taken for the two species (Cammeo and Osmancik), which were processed and feature inferences were made. 7 morphological features were obtained for each grain of rice which are the variables in the data set.

1. Divide the data randomly into training and test data (70/30). Assume that columns Area,…,Extent are denoted as $x_1, \dots, x_p$. Perform basis function expansion with basis functions $\phi_1 = x_1, \dots, \phi_p = x_p, \phi_{p+1} = x_1^2, \dots \phi_{2p} = x_p^2$ and fit two logistic regression models: one with features $x_1, \dots x_p$ and another with $\phi_1, \dots \phi_{2p}$, both with target Class to the training data. Report the training and test misclassification errors for both models and report which model is better and why. Finally, report the estimated probabilistic model for the model with features $x_1, \dots x_p$ **(4p)**
2. Use cross-validation to fit a decision tree model to the training data with target Class and all other columns $(x_1, \dots, x_p)$ as features and report the optimal number of leaves and training and test misclassification errors for the optimal tree. Comment on the prediction quality of the optimal tree. After this, write a loop that grows decision trees with parameter $mindev = 0.001, 0.002, \dots 0.01$ (without cross-validation) and estimates training and test misclassification errors for these trees. Plot the dependence of these errors on $mindev$, comment on the trends observed in the plot and comment how the trends are expected to look in theory and why. **(4p)**
3. Use the model estimated by the cross-validation in step 2 and assume that a user wants to prune this tree. If we use misclassification error as impurity measure, which leaves must be pruned first, 14 and 15, or 4 and 5? Report necessary mathematical calculations to support your answer. **(2p)**

Assignment 2 (10p)

NEURAL NETWORK – 10 POINTS

You are asked to implement the backpropagation algorithm for training a neural network for regression as it appears in the course textbook and slides. You can find the pseudocode below. The neural network has one hidden layer with two units. W denotes weights, b denotes intercepts, z denotes activation units, q denotes hidden units (i.e. the result of applying the activation function h to z), J denotes the squared error, and gamma denotes the learning rate. The superscript indicates the layer (0=input layer, 1=hidden layer, 2=output layer). All products are matrix products (%*% in R), except the one indicated with ⊙ that is element-wise product (* in R). Note the use of matrix transposition in some steps (t() in R). **Comment your code and results**. The exercise will be graded as follows: Forward propagation 2 p, backward propagation 4 p, parameter updating 1 p, and results 3 p.

- Forward propagation.

$$
\begin{aligned}
\boldsymbol{q}^{(0)} &= \boldsymbol{x} \\
\boldsymbol{z}^{(1)} &= \boldsymbol{W}^{(1)}\boldsymbol{q}^{(0)} + \boldsymbol{b}^{(1)} \\
\boldsymbol{q}^{(1)} &= h(\boldsymbol{z}^{(1)}) \\
z^{(2)} &= \boldsymbol{W}^{(2)}\boldsymbol{q}^{(1)} + b^{(2)} \\
J(\boldsymbol{\theta}) &= (y - z^{(2)})^2
\end{aligned}
$$

- Backward propagation.

$$
\begin{aligned}
dz^{(2)} &= -2(y - z^{(2)}) \\
d\boldsymbol{q}^{(1)} &= \boldsymbol{W}^{(2)T} dz^{(2)} \\
d\boldsymbol{z}^{(1)} &= d\boldsymbol{q}^{(1)} \odot h'(\boldsymbol{z}^{(1)}) \\
d\boldsymbol{W}^{(2)} &= dz^{(2)}\boldsymbol{q}^{(1)T} \\
db^{(2)} &= dz^{(2)} \\
d\boldsymbol{W}^{(1)} &= d\boldsymbol{z}^{(1)}\boldsymbol{q}^{(0)T} \\
d\boldsymbol{b}^{(1)} &= d\boldsymbol{z}^{(1)}
\end{aligned}
$$

- Parameter updating.

$$
\begin{aligned}
\boldsymbol{W}^{(2)}_{t+1} &= \boldsymbol{W}^{(2)}_{t} - \gamma d\boldsymbol{W}^{(2)}_{t} \\
\boldsymbol{b}^{(2)}_{t+1} &= \boldsymbol{b}^{(2)}_{t} - \gamma d\boldsymbol{b}^{(2)}_{t} \\
\boldsymbol{W}^{(1)}_{t+1} &= \boldsymbol{W}^{(1)}_{t} - \gamma d\boldsymbol{W}^{(1)}_{t} \\
b^{(1)}_{t+1} &= b^{(1)}_{t} - \gamma db^{(1)}_{t}
\end{aligned}
$$

You are requested to use the template below. Note that the algorithm performs 100000 iterations. In each iteration, one randomly selected training point is used to update the parameters (i.e., this essentially corresponds to stochastic gradient descent with a mini-batch of size 1).

```
# produce the training data in dat

x <- runif(500,-4,4)

y <- sin(x)

dat <- cbind(x,y)

plot(dat)

gamma <- 0.01

h <- function(z){

 # activation function (sigmoid)

 return(1/(1+exp(-z)))

}

hprime <- function(z){

 # derivative of the activation function (sigmoid)
```

```
  return(h(z) * (1 - h(z)))
}
yhat <- function(x){
  # prediction for point x
}
MSE <- function(){
  # mean squared error
}
# initialize parameters
res <- NULL
for(i in 1:100000){
  if(i %% 1000 == 0){
    res <- c(res,MSE())
  }
  # forward propagation
  j <- sample(1:nrow(dat),1)
  q0 <- dat[j,1]
  # backward propagation
  # parameter updating
}
plot(res, type = "l")
plot(dat)
points(dat[,1],lapply(dat[,1],yhat))
```