# Examination

**Linköping University, Department of Computer and Information Science, Statistics**

| | |
|---|---|
| Course code and name | TDDE01 Machine Learning |
| Date and time | 2024-08-28, 8.00-13.00 |
| Assisting teacher | Oleg Sysoev |
| Allowed aids | Contents of “help_materials” folder + 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 (allowed formats: 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)

Data file **Australian-crabs.csv** contains measurements about Australian crabs as well as their properties, such as sex or type of species.

1. Assume that Sex is the target variable in the modeling and all crab measurements are the features, and perform model selection by using decision trees and the hold-out method (60% train/ 40% test). Present a plot showing dependence of the training and test-cross entropies on the amount of leaves in the tree and report how many leaves the optimal tree has. Motivate your model choice. Why is cross-entropy a reasonable metric here? How many features are selected by the optimal tree? Why do tree branches in the tree with 7 leaves from some parent node lead to the leaves with exactly same labels? **(3p)**
2. Assume that Sex is the target variable in the modeling and all measurements are the features, and compute a logistic regression model by using the entire dataset. Compute the predicted probabilities for the first observation in the dataset. Use output from this model to compute how much these probabilities change if we set the parameters corresponding to CW and BD to zero. Report mathematical calculations that were used for computing the updated probabilities. Compute F1-score (positive class=Male) for the entire dataset by using the following loss matrix:

$$\begin{array}{c} \\ True\ "Male" \\ True\ "Female" \end{array} \begin{array}{c} \begin{array}{cc} Pred\ "Male" & Pred\ "Female" \end{array} \\ \begin{pmatrix} 0 & 1 \\ 10 & 0 \end{pmatrix} \end{array}$$

   Is F1 score or the accuracy a more relevant metric for this dataset? Motivate your answer. **(4p)**
3. Assume the following model that connects FL and RW ($w_0$ and $w_1$ are parameters:

$$RW \sim Normal(\mu = w_0 + w_1 FL, \sigma^2 = 0.1FL - 0.5)$$

   Implement a minus log-likelihood function in R that describes this model as a function of the parameters. Use the BFGS optimization method with starting point (0,0) to compute and report the optimal parameters, using the entire dataset. Finally, compute the prediction interval for the first observation in the dataset. **(3p)**

## Assignment 2 (10p)

NEURAL NETWORKS - 10 POINTS

Read the exercise entirely before starting. 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 $\odot$ that is element-wise product (* in R). Note the use of matrix transposition in some steps (t() in R). **Comment your code.**

- Forward propagation.

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

- Backward propagation.

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

- Parameter updating.

$$\boldsymbol{W}_{t+1}^{(2)} = \boldsymbol{W}_t^{(2)} - \gamma d\boldsymbol{W}_t^{(2)}$$
$$\boldsymbol{b}_{t+1}^{(2)} = \boldsymbol{b}_t^{(2)} - \gamma d\boldsymbol{b}_t^{(2)}$$
$$\boldsymbol{W}_{t+1}^{(1)} = \boldsymbol{W}_t^{(1)} - \gamma d\boldsymbol{W}_t^{(1)}$$
$$b_{t+1}^{(1)} = b_t^{(1)} - \gamma db_t^{(1)}$$

**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))
```

Finally, you are asked to **incorporate dropout** to your implementation of the backpropagation algorithm. Recall that dropout is a regularization technique whose detailed description you can find in the course textbook and slides. Run your implementation with a dropout rate 1-r equal to 0, 0.01 and 0.05, i.e. r=1, 0.99, 0.95. **Comment the results.**

The exercise will be graded as follows: Forward propagation 1 p, backward propagation 2 p, parameter updating 1 p, dropout 4 p, and results 2 p.