# Examination

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

| | |
|---|---|
| Course code and name | TDDE01 Machine Learning |
| Date and time | 2024-01-12, 14.00-19.00 |
| 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)

File **Bikes.csv** contains counts of public bicycles rented per hour in the Seoul Bike Sharing System, with corresponding weather data and holiday information.

1. Divide the data randomly into training and test data (70/30) and scale them appropriately. Compute a LASSO regression model by cross-validation in which Rented Bikes is the target variable and all remaining variables are features, and present a dependence of the cross-validation error on the penalty parameter. Which value of the penalty parameter is the optimal one? Interpret this model in terms of bias-variance tradeoff. Finally, report the equation showing how the predicted number of Rented bikes (scaled) depends on the features (scaled) in the estimated LASSO model corresponding to "$lambda.1se$" penalty value. **(4p)**
2. Consider $lambda.1se$ LASSO model, use the training error MSE as an estimate of the target variance parameter and report the 95% prediction interval for the first observation (first row) in the test data. Explain the model assumptions making these computations possible. **(2p)**
3. Consider the same partitioned and scaled data as in step 1. Assume now that Dew Point Temperature is related to the features Humidity and Visibility as a linear model without intercept, and assume that the loss and the error functions are given by the following formula:
$$L(y, \hat{y}) = E(y, \hat{y}) = |y - \hat{y}|$$
Implement a code optimizing the cost function by the BFGS optimizer, and plot a dependence of the cost values and the test errors on the iteration number. Is early stopping needed? Report the optimal iteration number. Make 3 scatter plots of (Humidity,Visibility) where observations are colored by a) original target values b) predicted target values from the optimal model c) target values from the model corresponding to 5 iterations, and compare the plots with respect to the complexity of the model and quality of prediction. **(4p)**
    - Hint: to make scatter plots, you may use this kind of code from package **ggplot2**:
```
df=data.frame(x=your_variable1, y=your_variable2, color=your_variable3)
ggplot(df,aes(x=x,y=y,color=color))+geom_point()
```

## Assignment 2 (10p)

EXERCISE 1 – **5 POINTS**

In January 2023, the students were asked to implement the backpropagation algorithm for training a neural network for regression as it appears in the course textbook and slides. The solution is available to you in the file TDDE01January2023.R. Now, you are asked to incorporate dropout to this solution. 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.

EXERCISE 2 – **5 POINTS**

You are asked to implement the perceptron algorithm. This algorithm for binary classification is a predecessor of modern neural networks.

Consider a binary classification problem with class labels $t \in \{-1, +1\}$. Then, the class label assigned to a point $x$ is given by

$$y(\boldsymbol{w}, \boldsymbol{x}) = \begin{cases} +1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} \geq 0 \\ -1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} < 0. \end{cases}$$

The values of $w$ are iteratively determined with the help of the learning data $\{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$. Specifically, the $i$-th weight is updated in each iteration as follows:

$$\boldsymbol{w}_i^{(new)} = \boldsymbol{w}_i^{(old)} + \alpha \sum_{n=1}^{N} (t_n - y(\boldsymbol{w}^{(old)}, \boldsymbol{x}_n))\boldsymbol{x}_{n,i}$$

where $\alpha$ is the learning rate. Finally, you can assume that $\boldsymbol{x}_n$ and $\boldsymbol{w}$ are of dimension two, i.e. $\boldsymbol{x}_n = (\boldsymbol{x}_{n,1}, \boldsymbol{x}_{n,2})$ and $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2)$.

You can stop the learning process after 100 iterations. You can use an alpha value of 0.0001. Plot the the misclassification rate on the dataset below as a function of the number of iterations.

```
set.seed(1234)

x <- array(NA, dim = c(100,2))

t <- array(NA, dim = c(100))

x[,1] <- runif(100,0,3)

x[,2] <- runif(100,0,9)

t <- ifelse(x[,2]<(x[,1])^2,-1,1)

plot(x[,1],x[,2],col=t+2)
```

Finally, explain when the perceptron algorithm works best and why it works in those cases.