دانشگاه صنعتی شریف	**مقدمه ای بر یادگیری ماشین ۲۵۷۳۷**
دانشکده مهندسی برق	**گروه ۱**
نیمسال پاییز ۱۴۰۱-۱۴۰۰	**مدرس: سید جمال الدین گلستانی**

تکلیف کامپیوتری شماره 2

موعد تحویل: جمعه 21 آبان 1400

توضیحات کلی

- تمام فایل‌های مربوط به سوالات کامپیوتری را در یک فایل به نام CHW2N.zip قرار دهید که N شماره دانشجویی شماست.
- سوالات خود را در مورد این تکلیف با دستیار آموزشی آقای امیررضا وظیفه در آدرس ایمیل amirrezavazifeh2000@hotmail.com مطرح کنید.

در پایان فایل های نوت بوک به فرمت ipynb. را که هم شامل کدها و نتایج و هم شامل گزارش هست بفرستید. سعی کنید تمام چیزهایی که خواسته شده را داخل نوت بوک ها بنویسید اما اگر راحت تر بودید که بعضی سوالات تشریحی را به دلیل نیاز به فرمول نویسی یا موارد دیگر در Word یا ... بنویسید، می توانید این کار را انجام دهید اما در همان فایل نوت بوک بگویید که در کجا پاسخ این قسمت داده شده است.

Problem C3: Linear Classification & Decision Trees

In this problem you will implement linear classification algorithms (perceptron and linear support vector machines), and decision trees. Answer the questions in your report, which should not exceed

three pages.

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

A. Load Data

In this section, you are given a data set data_banknote_authentication.csv with two classes (y = 1, y = -1). Each data point has four features obtained from digital image processing of fake and real banknotes, and a label y = 1 for real backnotes, and y = -1 for faked ones. The first four columns are training features and the last column is the label.

Split the data set into two parts. The first 80% of the data is for training and the remaining 20% for testing. Import data with pandas library. The first 4 columns are training features, denoted by x_1; x_2; x_3; x_4, and the 5th column is the label, denoted by y.

[8]:
```
## your code to load data

# X = ...
# Y = ...



# Your code to split the data (with no randomization)


# X_train = ...
# Y_train = ...


# X_test = ...
# Y_test = ...
```

B. Perceptron Algorithm

In this part, you should implement the perceptron algorithm from the scratch. First, add one dimension with constant 1 to each data point (i.e. x_0 = 1). What is the purpose of this? Perform the algorithm for 50000 iterations, and at each 500 iterations calculate and save the error $(\frac{1}{n}\sum_{i=1}^{n} 1_{\text{prediction}_i \neq y_i})$ on the test data, and in the end plot the error against the number of iterations. After the end of the training process, report the final error on the test data and the final weights w.

[1]:
```
## Implement the algorithm here

def perceptron(X_tr, Y_tr, X_te, Y_te, max_iter=50000):

    w = 0
    loss_history = list()


    ################################
    ################################

            # Your code


    ################################
    ################################


    return w, loss_history

## train of training samples

# w, loss_history = perceptron(X_train, Y_train, X_test, Y_test)

final_loss = 0
```

[2]:
```
## Plot test error over every 500 iterations
```

C. Generalize to non-linear classification

Map data (train and test) x by $\psi(.)$ to the six-dimensional x' as follows:

$$x = (x_0, x_1, x_2, x_3, x_4) \xrightarrow{\psi} x' = (x_0, x_1, x_2, x_3, x_4, x_4^3)$$

Apply Perceptron to x' and repeat part A2.

[3]:
```
## your code to train on x'
```

[4]:
```
## plot error on x'
```

D. SVM algorithm

Train the SVM model on the data. In this part of the problem, you should use built-in models of libraries like Sklearn for training and predicting labels of the data. Do not change the default parameters of the model excpet *max_iter* if it is needed. Note that you should train the model on the pure form of x (without the added feature $x_0 = 1$). At last, report final error $(\frac{1}{n}\sum_{i=1}^{n} 1_{\text{prediction}_i \neq y_i})$ on the training and the test samples, and final weights.

[5]:
```
## Train a SVM model, report final errors

emp_loss, true_loss = 0, 0
weights = 0
```

E. Conclusion

discuss and compare the resulting weights and errors ($\frac{1}{n}\sum_{i=1}^{n} 1_{\text{prediction}_i \neq y_i}$) of the above methods in your report.

Problem C4: Decision Trees

The dataset mushrooms.csv includes the overall features of some population of mushrooms. Each data point has 22 features (e.g. habitat, size, color, etc.) and the goal is to classify mushrooms as poisonous ($y = 0$) or edible ($y = 1$), by using Decision Trees classifiers. Use built-in models of libraries in Sklearn for training and predicting labels of the data. Do not change the default parameters of the model excpet *max_depth*.

A. Load Data

The first cloumn is the label, and the remaining 22 columns are features of data points. Split the data into three sets: the first 70% for training, the next 20% for verification, and the remaining 10% for testing. The validation set is for choosing the best model among all models based on the error ($\frac{1}{n}\sum_{i=1}^{n} 1_{\text{prediction}_i \neq y_i}$). The test set is for estimating the true error of the selected model. Import data with pandas library.

[9]:
```
## your code to load data

# X = ...
# Y = ...



# Your code to split the data (with no randomization)


# X_train = ...
# Y_train = ...


# X_val = ...
# Y_val = ...


# X_test = ...
# Y_test = ...
```

B. Train Desicion Tree

Set the maximum depth of the tree to *{4; 6; 8; 10; 12; 14; 16; 18; 20}*. For each maximum depth, train
a classifier on the training data and report the resulting loss on the validation set. Plot the loss against the maximum depth. What is the best maximum depth? Finally, for the best maximum depth, report the loss on the test set.

[10]:
```
## Train Desicion Tree for each depth here

depth_tree = [4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Plot validation errors over the depth of trees, and explain your observations.

[13]:
```
## Implement the figure here
```