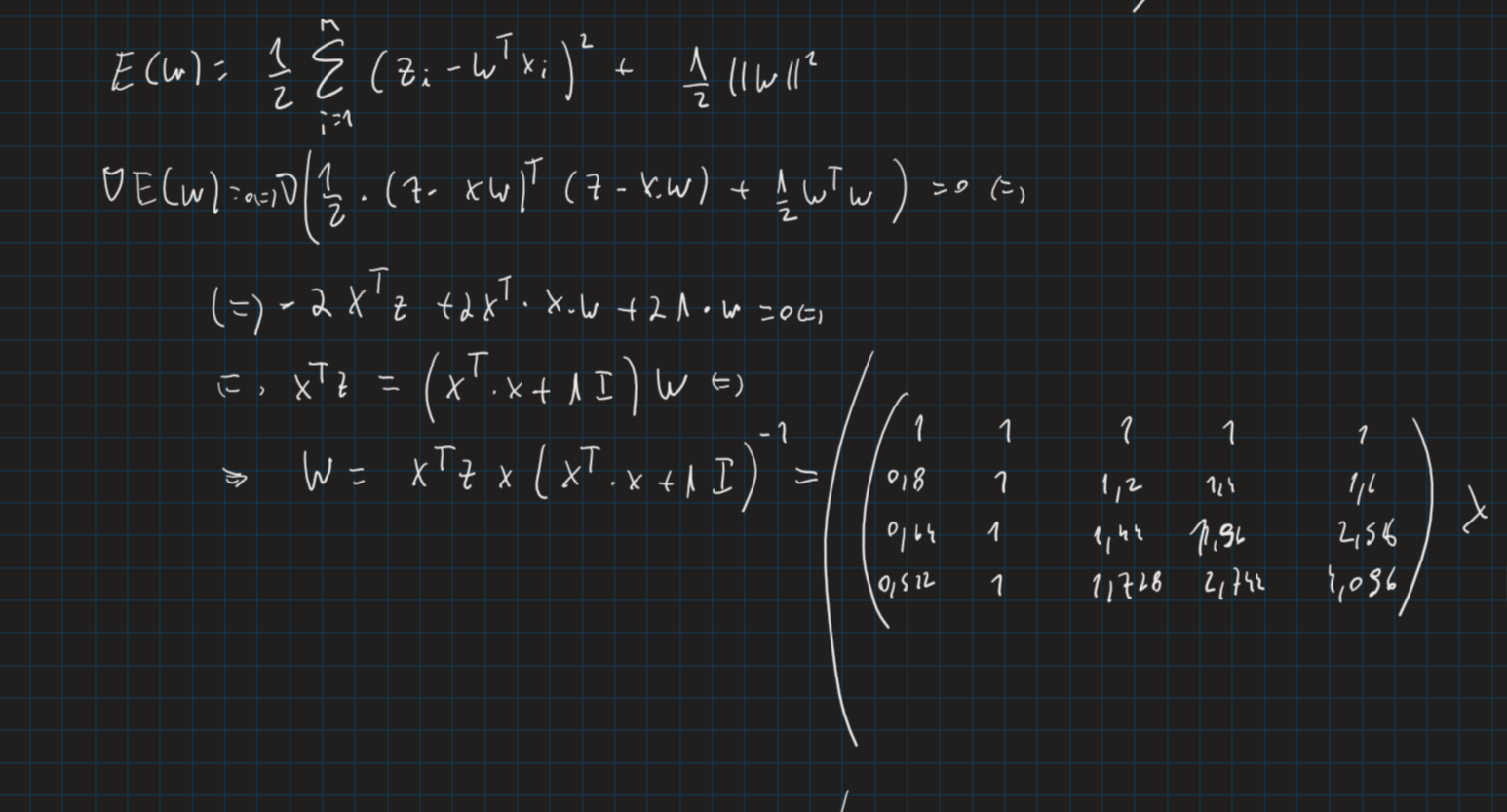
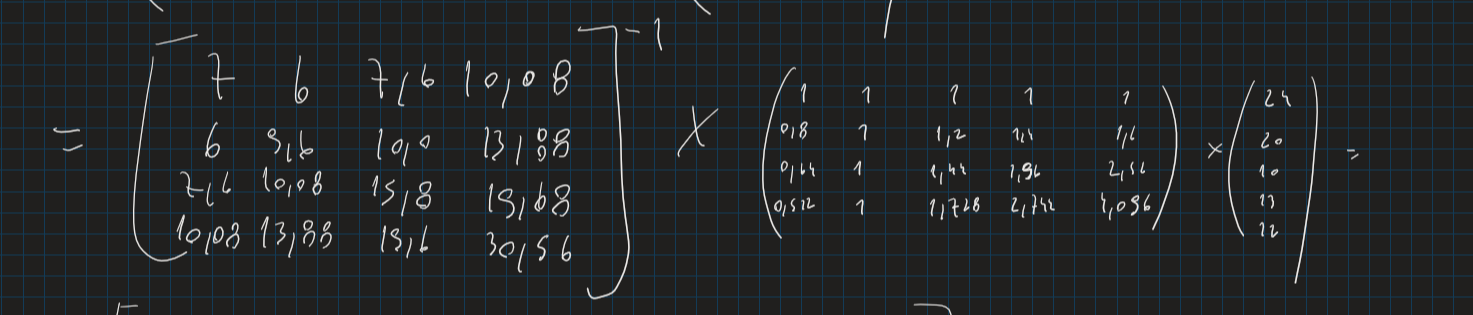
**I. Pen-and-paper**

1. Uma imagem com texto

   Descrição gerada automaticamente

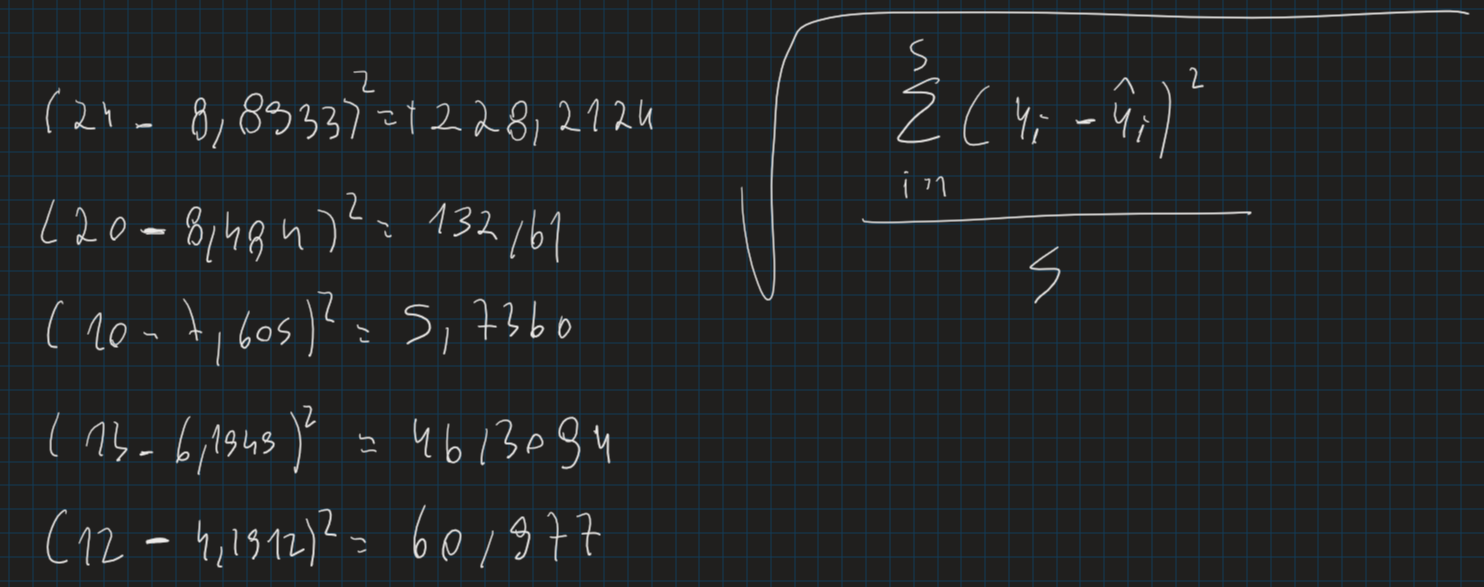
Uma imagem com texto, eletrónica, em mosaico

Descrição gerada automaticamente

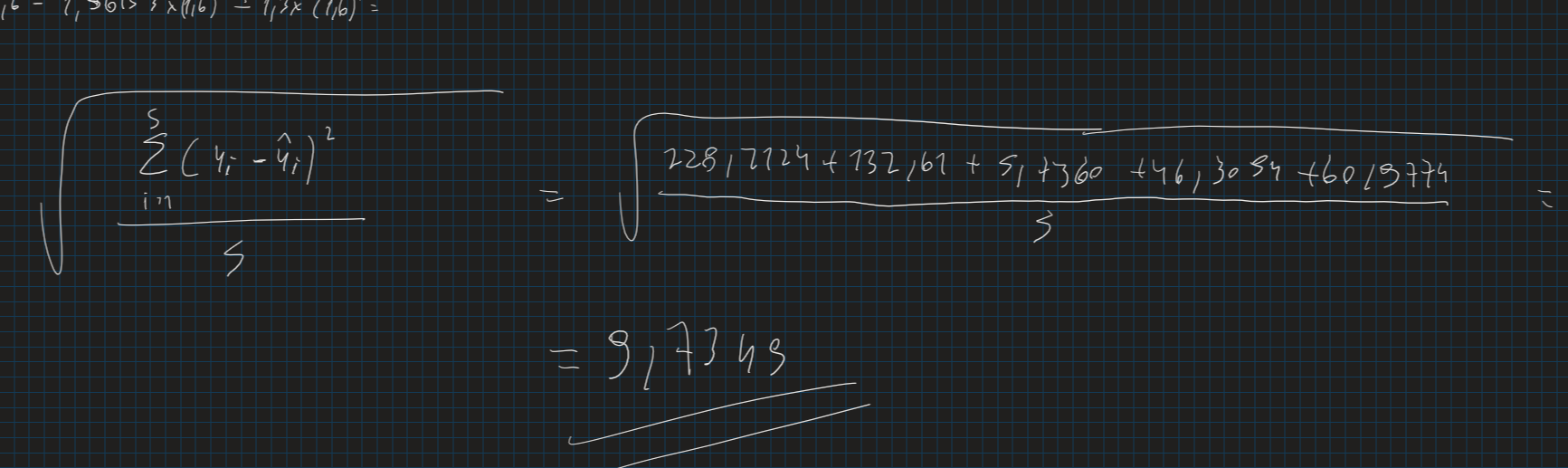


Uma imagem com texto, casa de banho, em mosaico

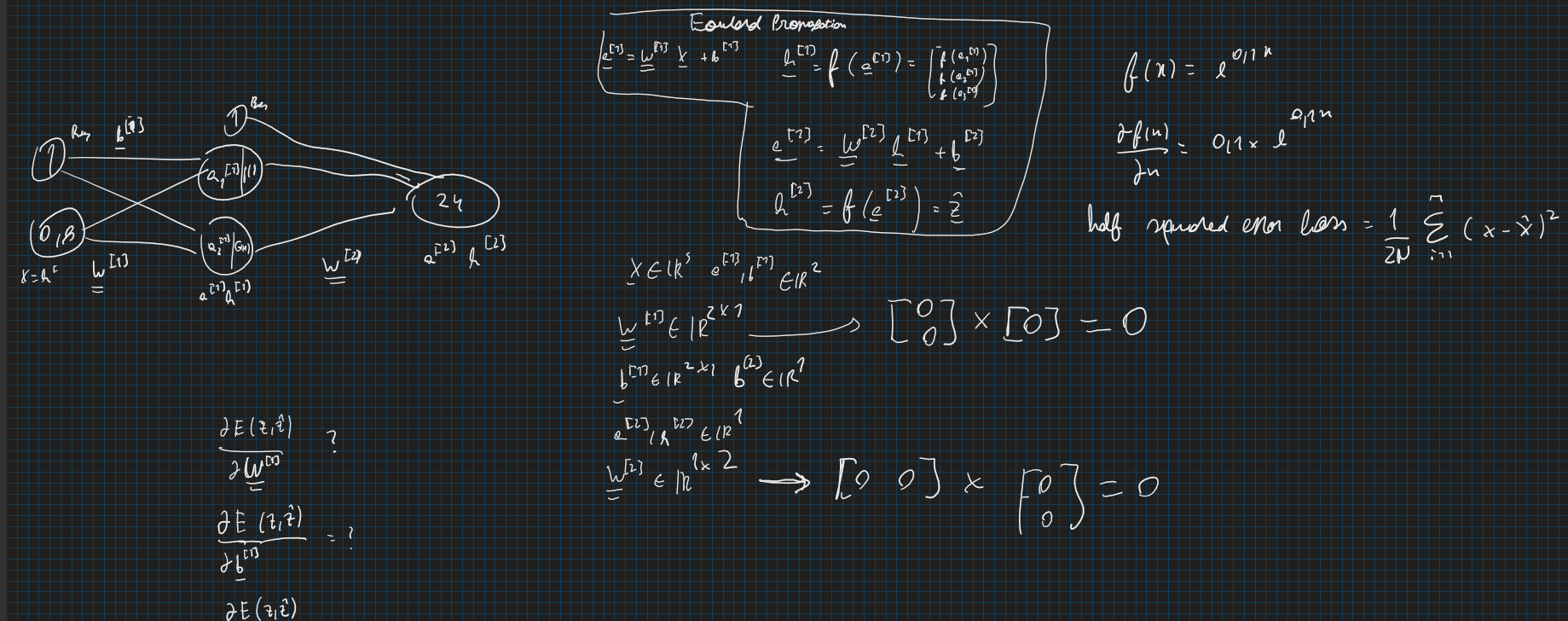
Descrição gerada automaticamente

1. Uma imagem com texto, quadro-preto

   Descrição gerada automaticamente



1. Uma imagem com texto, eletrónica

   Descrição gerada automaticamente

Uma imagem com texto, quadro-preto

Descrição gerada automaticamenteUma imagem com texto, eletrónica, circuito

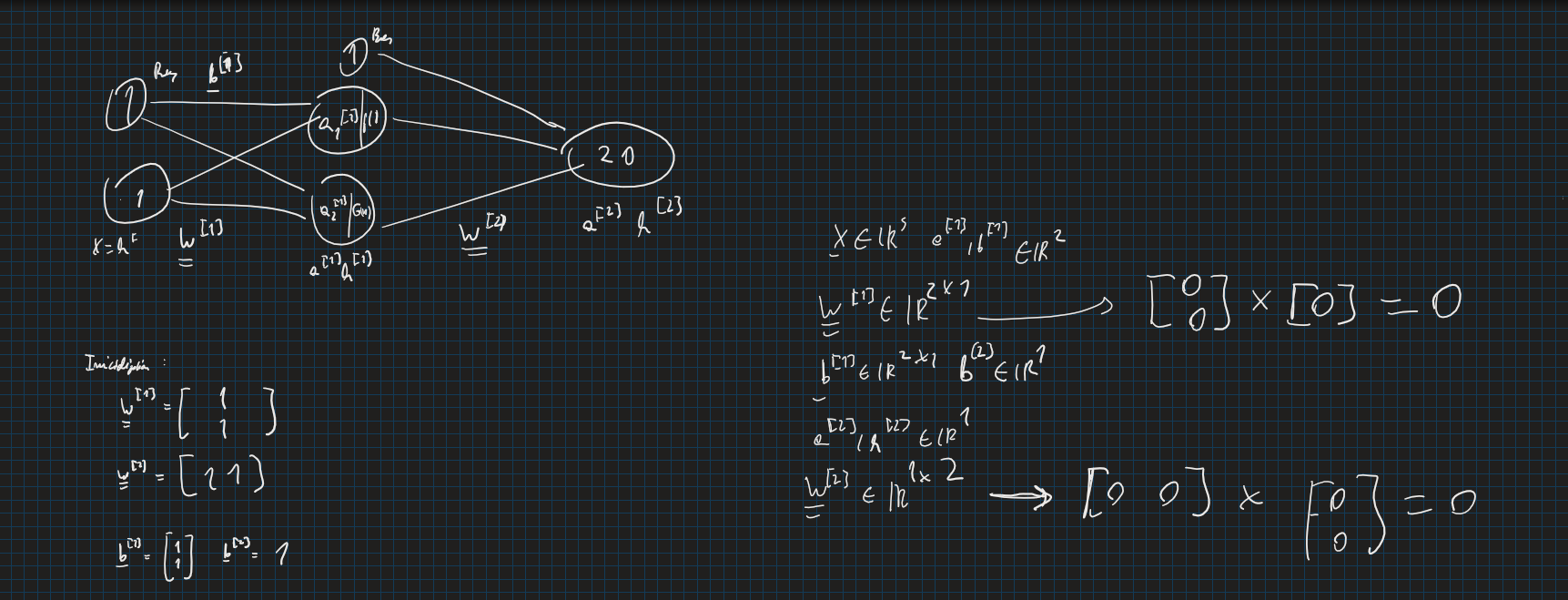
Descrição gerada automaticamente

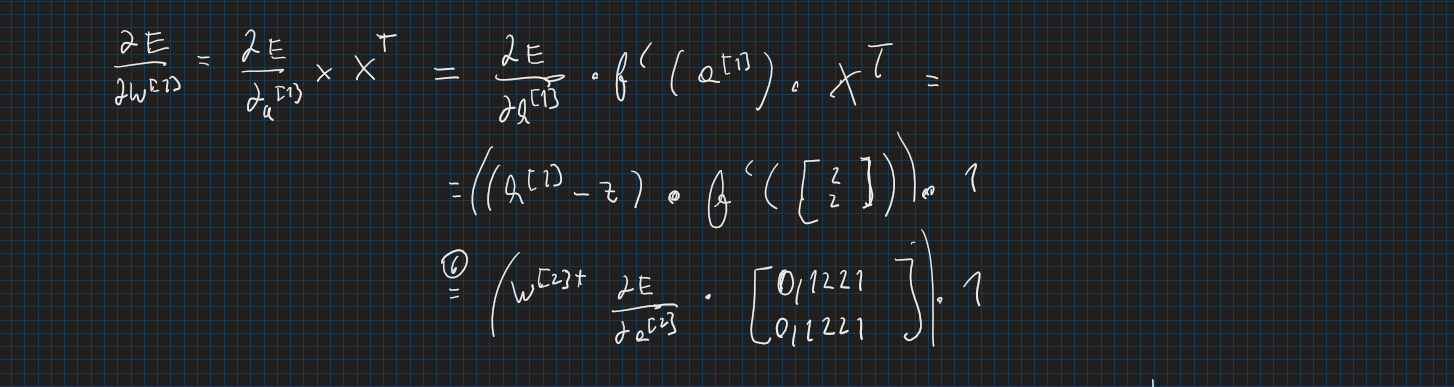
Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto, interior, janela, eletrónica

Descrição gerada automaticamente



Uma imagem com texto, quadro-preto

Descrição gerada automaticamenteUma imagem com texto, eletrónica, circuito

Descrição gerada automaticamente

Uma imagem com texto, janela, eletrónica

Descrição gerada automaticamenteUma imagem com texto, eletrónica, circuito

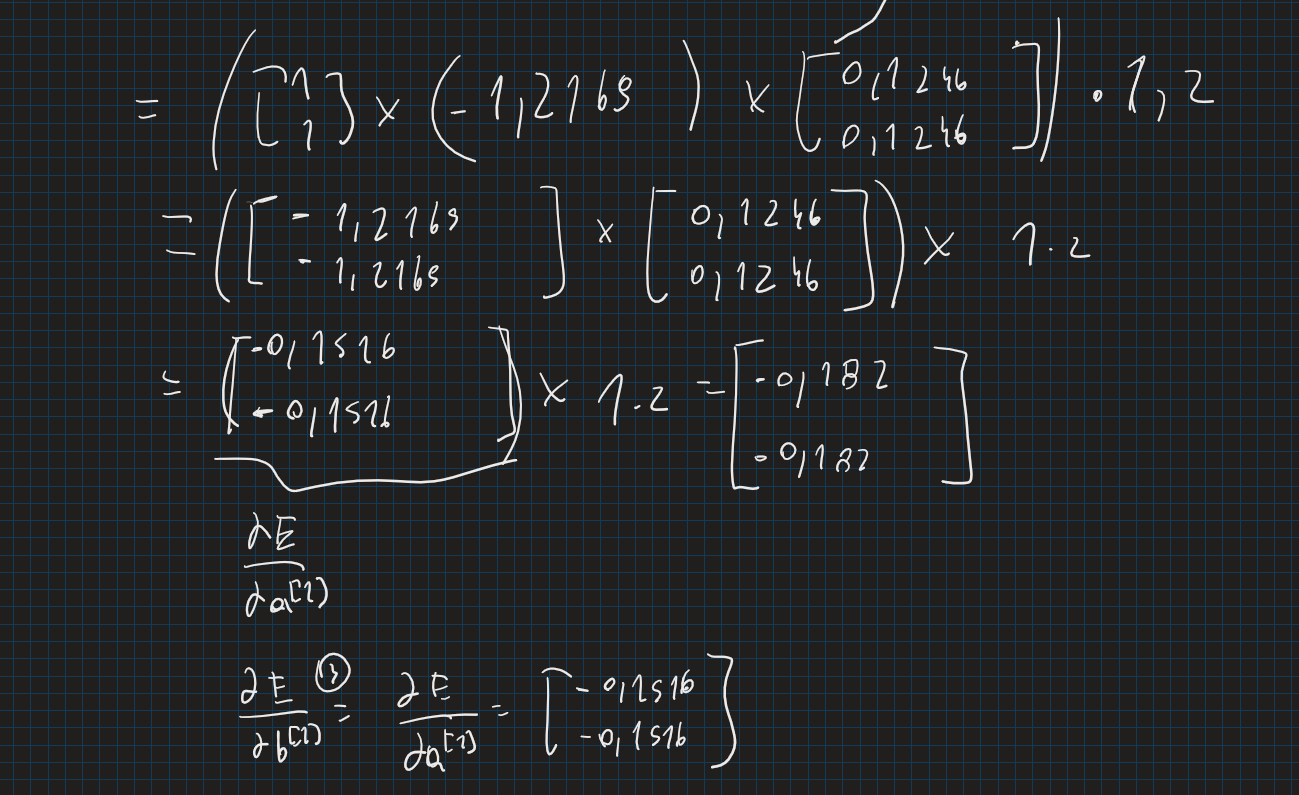
Descrição gerada automaticamenteUma imagem com texto, interior, em mosaico

Descrição gerada automaticamente



Uma imagem com texto, eletrónica, circuito

Descrição gerada automaticamente



Uma imagem com texto, eletrónica, em mosaico

Descrição gerada automaticamenteUma imagem com texto, janela

Descrição gerada automaticamente

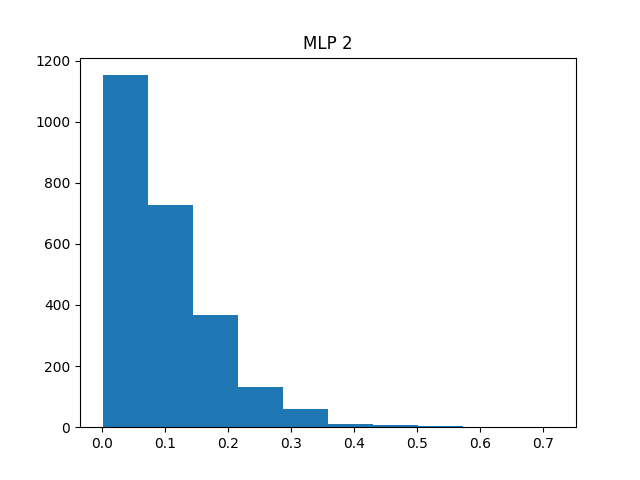
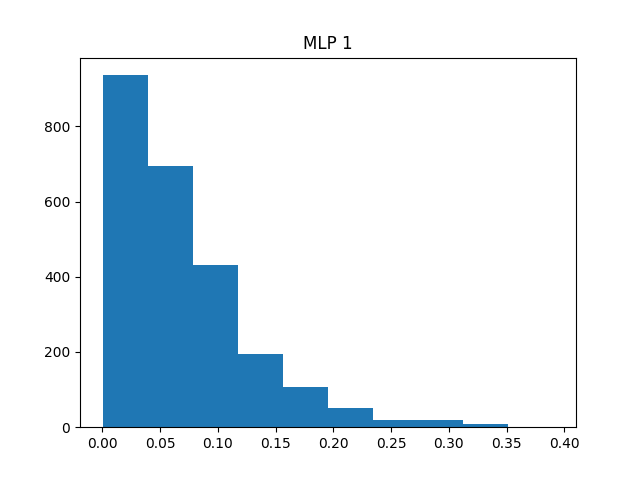
**II. Programming and critical analysis**

1. Ridge MAE: 0.162829976437694  
   MLP 1 MAE: 0.0680414073796843  
   MLP 2 MAE: 0.0978071820387748
2. Chart, box and whisker chart

   Description automatically generatedChart, box and whisker chart

   Description automatically generated  
   Chart, box and whisker chart

   Description automatically generatedChart, histogram

   Description automatically generated  
     
   
3. Iterações do MLP 1: 452  
   Iterações do MLP 2: 77
4. As iterações do MLP 1 são maiores que as do MLP 2 o que é o contrário do esperado. Tem-se que quando o modelo aprende com early restart, como é o caso do MLP 1, o processo de aprendizagem é parado apenas quando não é detetado um melhoramento na pontuação de validação igual ou superior a um determinado valor durante um determinado número de iterações. Isto é feito para contornar o potencial barulho estatístico no dataset. Tendo isto em consideração, a presença de bastante barulho estatístico no dataset pode motivar o maior número de iterações em MLP 1, uma vez que a paragem do processo é adiada várias vezes.   
   Quanto à diferença de performance, o MLP 1 resultou num modelo que não está over-fitted, o que resulta numa melhor performance comparativamente ao MLP 2.

**III. APPENDIX**

Paste your programming code here using Consolas 9pt or 10pt.

Use **highlighting** or colored text to facilitate the analysis by your faculty hosts.

import pandas as pd

import numpy as np

#\*###################################################################################

#\* 4)

#\*###################################################################################

#\* import data

from scipy.io.arff import loadarff

data = loadarff("kin8nm.arff")

df = pd.DataFrame(data[0])

num\_columns = df.shape[1]

#\* partition data

from sklearn.model\_selection import train\_test\_split

X, y = df.iloc[:, 0:num\_columns-1], df['y']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size = 0.70, random\_state=0)

#\* linear regressor with Ridge regularization

from sklearn.linear\_model import Ridge

rr = Ridge(alpha=0.1)

#\* MLP\_1 and MLP\_2 regressors

from sklearn.neural\_network import MLPRegressor

MLP1r = MLPRegressor(hidden\_layer\_sizes=(10, 10), activation="tanh", max\_iter=500, random\_state=0, early\_stopping=True)

MLP2r = MLPRegressor(hidden\_layer\_sizes=(10, 10), activation="tanh", max\_iter=500, random\_state=0, early\_stopping=False)

#\* learn (".values" was added to avoid warnings)

rr.fit(X\_train.values, y\_train.values)

MLP1r.fit(X\_train.values, y\_train.values)

MLP2r.fit(X\_train.values, y\_train.values)

#\* predict (".values" was added to avoid warnings)

rr\_y\_pred = rr.predict(X\_test.values)

MLP1r\_y\_pred = MLP1r.predict(X\_test.values)

MLP2r\_y\_pred = MLP2r.predict(X\_test.values)

#\* compute MAE

from sklearn.metrics import mean\_absolute\_error

y\_true = y\_test

rr\_MAE = mean\_absolute\_error(y\_true, rr\_y\_pred)

MLP1r\_MAE = mean\_absolute\_error(y\_true, MLP1r\_y\_pred)

MLP2r\_MAE = mean\_absolute\_error(y\_true, MLP2r\_y\_pred)

print("Ridge MAE: " + str(rr\_MAE))

print("MLP1 MAE: " + str(MLP1r\_MAE))

print("MLP2 MAE: " + str(MLP2r\_MAE))

#\*###################################################################################

#\* 5)

#\*###################################################################################

#\* array of residuals (in absolute value)

rr\_residuals = abs(y\_true - rr\_y\_pred)

MLP1r\_residuals = abs(y\_true - MLP1r\_y\_pred)

MLP2r\_residuals = abs(y\_true - MLP2r\_y\_pred)

#\* plot

import matplotlib.pyplot as plt

plt.boxplot(x=rr\_residuals)

plt.title(label="Ridge")

plt.show()

plt.boxplot(x=MLP1r\_residuals)

plt.title(label="MLP 1")

plt.show()

plt.boxplot(x=MLP2r\_residuals)

plt.title(label="MLP 2")

plt.show()

plt.hist(x=rr\_residuals)

plt.title(label="Ridge")

plt.show()

plt.hist(x=MLP1r\_residuals)

plt.title(label="MLP 1")

plt.show()

plt.hist(x=MLP2r\_residuals)

plt.title(label="MLP 2")

plt.show()

#\*###################################################################################

#\* 6)

#\*###################################################################################

MLP1r\_iterations = MLP1r.n\_iter\_

MLP2r\_iterations = MLP2r.n\_iter\_

print('MLP1 Iterations: ' + str(MLP1r\_iterations))

print('MLP2 Iterations: ' + str(MLP2r\_iterations))

**END**