

```

# -*- coding: utf-8 -*-
"""
Created on Wed Dec 1 11:22:47 2021

@author: Nanto RALIBERA
@modified by : QUESNEL Kévin-Lâm
"""

import numpy as np
import random as rd
from tqdm import tqdm
import os

#LOAD DATA
#building the right_matrix

right_matrix=np.load("right_matrix.npy")

#build permanent fault
matrix_data = np.load("permanent_fault_array.npy")
label = np.load("fault_label.npy")
start=int(input("what class do we start from ? ")) #if there is an interruption when generating the faults

#build intermittent fault
#activation_rate = 0.85 #85% des vecteurs de la faute permanente restent intacts le reste est remplacés
nb_of_intermittent_fault=100
nb_activ_rate=30

for i in tqdm(range(start,matrix_data.shape[2])): #on fait des fautes intermittentes pour chaque classe de fautes

    faulty_line=[]

    final_matrix = np.copy(matrix_data[:, :, i])
    final_matrix = np.reshape(final_matrix, (np.shape(matrix_data)[0], np.shape(matrix_data)[1], 1))

    final_label = np.array([label[i]])

    matrix_copy= np.copy(matrix_data[:, :, i])
    matrix_copy= np.reshape(matrix_copy, (np.shape(matrix_data)[0], np.shape(matrix_data)[1], 1))

    for j in range(matrix_data.shape[0]): #get the faulty lines
        for k in range(matrix_data.shape[1]):
            if matrix_copy[j,k] != right_matrix[j,k,0]:
                faulty_line.append(j)
                break
    #print('number of faulty line =', len(faulty_line))

    for k in range(nb_activ_rate+1): #no need for an activation rate = 1 it's just
        #permanent faults

        activation_rate=k/nb_activ_rate#nb_activ_rate we wanna generate
        #print('k=', k)
        #print('activation_rate=', activation_rate)
        path="/home/projet12/PAr135/intermittent_fault/activation_rate"+str(k)+"sur"+str(nb_activ_rate)
        #i don't know why but the server console was printing int(activation_rate)
        if not os.path.exists(path):
            os.makedirs(path) #create a new file for this specific activation rate
        else:
            os.chdir(path)
        #change faulty line
        list_of_sample = []
        for index in range(nb_of_intermittent_fault):
            nb_of_vect_to_change=int((1-activation_rate)*len(faulty_line))
            sample = rd.sample(faulty_line, nb_of_vect_to_change)

            if sample not in list_of_sample:
                list_of_sample.append(sample)

        for elem1 in list_of_sample:
            for elem2 in range(len(elem1)):
                #print(elem1[elem2])
                matrix_copy[elem1[elem2], :, 0] = right_matrix[elem1[elem2], :, 0]

```

```
final_label=np.concatenate([final_label,np.array([i])]) #change faulty line selected into right ones
final_matrix=np.concatenate([final_matrix,matrix_copy],axis=2)

matrix_copy= np.copy(matrix_data[:, :, i])
matrix_copy= np.reshape(matrix_copy,(np.shape(matrix_data)[0],np.shape(matrix_data)[1],1))
#for a specific class and for a specific activation rate we have generated the matrix with
#the corresponding intermittent fault

filename="intermittent_fault_array"+str(i)

os.chdir(path)
np.savez_compressed(filename,matrix=final_matrix,label=final_label)
```