```
import keras
import numpy as np
from bdm import BDM
import random
from keras.preprocessing.sequence import pad_sequences

# Initialize BDM object
# ndim argument specifies dimensionality of BDM
bdm = BDM (ndim = 1)

length_max = 100
length_min = 21
num = 30000 #number of sequences to generate
X = [] #Training vector
Y = np.zeros ((num + 1, 1)) #Target Vector


for i in range (num + 1) :
    length = random.randint (length_min, length_max) #length of the sequence
    ones_prob = random.randint(0, 100)/100 #probability parameter
    if ones_prob > 0.95 : #increase the probability of a sequence full of 1s
        sec = np.random.choice (2, length, p = [0, 1]).reshape (1, length) #random sequence
        X.append (sec[0].tolist ())
        Y[i] = bdm.bdm (sec[0])
    else :
        prob = random.randint (0, 100)/100 #probability parameter
        sec = np.random.choice (2, length, p = [1 - prob, prob]).reshape (1, length)
        X.append (sec[0].tolist ())
        Y[i] = bdm.bdm (sec[0])

X = pad_sequences (X, value = -1) #padding procedure
```