class Hebb:  
 #конструктор  
 def \_\_init\_\_(self**,** inputs**,** outputs):  
 self.outputs = outputs  
 self.inputs = [[**1** for i in range(len(outputs))]] + inputs  
 self.weights = self.get\_weights()  
  
 #транспонування вектора  
 def transpose(self**,** vect):  
 return [**1** / vect[i] for i in range(len(vect))]  
  
 #скалярний добуток  
 def scalar(self**,** f**,** s):  
 return sum(f[i] \* s[i] for i in range(len(f)))  
  
 #знаходження вагів  
 def get\_weights(self):  
 weights = []  
 transposed\_vectors = list(map(self.transpose**,** self.inputs))  
 transposed\_result = self.transpose(self.outputs)  
 for i in range(len(transposed\_vectors)):  
 weights.append((self.scalar(transposed\_result**,** transposed\_vectors[i])) / len(self.outputs))  
 return weights  
  
 #порогова функція  
 def treshold\_function(self**,** arr):  
 new\_arr = []  
 for i in range(len(arr)):  
 if arr[i] > **0**:  
 new\_arr.append(**1**)  
 else:  
 new\_arr.append(-**1**)  
  
 return new\_arr  
  
 #знаходження зваженої суми  
 def get\_weighted\_sum(self):  
 arr = []  
 new\_array = list(zip(\*self.inputs))  
 for i in range(len(self.outputs)):  
 temp = self.scalar(self.weights**,** new\_array[i])  
 arr.append(temp)  
 return arr  
  
 #фінальний результат  
 def get\_result(self):  
 arr = self.get\_weighted\_sum()  
 return self.treshold\_function(arr)  
  
#перехід до іншого алфавіту {0,1} -> {1,-1}  
def to\_alphabet(number):  
 temp = bin(number)  
 temp = list(map(int**,** temp[**2**:]))  
 if len(temp) != **4**:  
 temp = [**0**] \* (**4** - len(temp)) + temp  
 for i in range(len(temp)):  
 if temp[i] == **0**:  
 temp[i] = **1** else:  
 temp[i] = -**1** return temp  
  
print("--------------------------------------Homework 1-------------------------------------------")  
  
x1 = [**1,1,**-**1,**-**1**]  
x2 = [**1,**-**1, 1,**-**1**]  
result = [to\_alphabet(i) for i in range(**16**)]  
  
for i in range(len(result)):  
 neiron = Hebb([x1**,**x2]**,** result[i])  
 print("Inputs: {0}, output: {1}".format(neiron.inputs[**1**:]**,** neiron.outputs))  
 print("Weights: {0}".format(neiron.weights))  
 print("Result: {0}".format(neiron.get\_result()))  
 if neiron.get\_result() == result[i]:  
 print("Correct")  
 else:  
 print("Wrong on {0} set".format(result[i]))  
 print("-"\***75**)  
  
print("----------------------------------------------------------Homework 2 Ex 1 Lect 3 №1-----------------------------------------------------------")  
  
x1 = [**1,1,1,1,**-**1,**-**1,**-**1,**-**1**]  
x2 = [**1,1,**-**1,**-**1,1,1,**-**1,**-**1**]  
x3 = [**1,**-**1,1,**-**1,1,**-**1,1,**-**1**]  
result = [**1,1,1,**-**1,1,1,**-**1,**-**1**]  
neiron = Hebb([x1**,**x2**,**x3]**,** result)  
print("Inputs: {0}, output: {1}".format(neiron.inputs[**1**:]**,** neiron.outputs))  
print("Weights: {0}".format(neiron.weights))  
print("Result: {0}".format(neiron.get\_result()))  
if neiron.get\_result() == result:  
 print("Correct")  
else:  
 print("Incorrect")  
  
print("-----------------------------------------------------------------Homework 2 Ex 1 Lect 3 №2----------------------------------------------")  
  
x1 = [**1,1,1,1,**-**1,**-**1,**-**1,**-**1**]  
x2 = [**1,1,**-**1,**-**1,1,1,**-**1,**-**1**]  
x3 = [**1,**-**1,1,**-**1,1,**-**1,1,**-**1**]  
result = [**1,**-**1,**-**1,**-**1,1,**-**1,1,**-**1**]  
neiron = Hebb([x1**,**x2**,**x3]**,** result)  
print("Inputs: {0}, output: {1}".format(neiron.inputs[**1**:]**,** neiron.outputs))  
print("Weights: {0}".format(neiron.weights))  
print("Result: {0}".format(neiron.get\_result()))  
if neiron.get\_result() == result:  
 print("Correct")  
else:  
 print("Incorrect")  
  
print("-------------------------------------------------------------Homework 2 Ex 2------------------------------------------------------------")  
  
x1 = [**0.5,0.5,0.4,0.4,**-**0.3,**-**0.3,**-**0.7,0.7**]  
x2 = [**1,1,**-**0.5,**-**0.5,0.7,0.7,**-**1,**-**1**]  
x3 = [**0.5,**-**0.3,0.4,**-**0.5,0.5,**-**0.4,0.3,**-**0.5**]  
result = [[**1,1,1,**-**1,1,1,**-**1,**-**1**]**,** [**1,**-**1,**-**1,**-**1,1,**-**1,1,**-**1**]]  
for i in range(len(result)):  
 neiron = Hebb([x1**,** x2**,** x3]**,** result[i])  
 print("Inputs: {0}, output: {1}".format(neiron.inputs[**1**:]**,** neiron.outputs))  
 print("Weights: {0}".format(neiron.weights))  
 print("Result: {0}".format(neiron.get\_result()))  
 if neiron.get\_result() == result[i]:  
 print("Correct")  
 else:  
 print("Wrong on {0} set".format(result[i]))  
 print("-" \* **175**)









