zadanie

June 20, 2025

[ ]:

```
import itertools
import numpy as np
import matplotlib.pyplot as plt


np.random.seed(1)

def printSample(x1, x2, t, y=None):
    x1 = ''.join([str(int(d)) for d in x1])
    x1_r = int(''.join(reversed(x1)), 2)
    x2 = ''.join([str(int(d)) for d in x2])
    x2_r = int(''.join(reversed(x2)), 2)
    t = ''.join([str(int(d[0])) for d in t])
    t_r = int(''.join(reversed(t)), 2)
    if y is not None:
        y = ''.join([str(int(d[0])) for d in y])
    print(f'x1:   {x1:s}   {x1_r:4d}')
    print(f'x2: + {x2:s}   {x2_r:4d}')
    print(f'      -------   ----')
    print(f't:  = {t:s}   {t_r:4d}')
    if y is not None:
        print(f'y:  = {y:s}')

def create_sum_dataset(nb_samples, sequence_len):
    max_int = 2**(sequence_len-1)
    format_str = '{:0' + str(sequence_len) + 'b}'
    X = np.zeros((nb_samples, sequence_len, 2))
    T = np.zeros((nb_samples, sequence_len, 1))
    for i in range(nb_samples):
        nb1 = np.random.randint(0, max_int)
        nb2 = np.random.randint(0, max_int)
        X[i,:,0] = list(reversed([int(b) for b in format_str.format(nb1)]))
        X[i,:,1] = list(reversed([int(b) for b in format_str.format(nb2)]))
        T[i,:,0] = list(reversed([int(b) for b in format_str.format(nb1 +␣
 ↪nb2)]))
    return X, T
```

```
def create_sub_dataset(nb_samples, sequence_len):
    max_int = 2**(sequence_len-1)
    format_str = '{:0' + str(sequence_len) + 'b}'
    X = np.zeros((nb_samples, sequence_len, 2))
    T = np.zeros((nb_samples, sequence_len, 1))
    for i in range(nb_samples):
        nb1 = np.random.randint(0, max_int)
        nb2 = np.random.randint(0, max_int)
        nb1, nb2 = max(nb1, nb2), min(nb1, nb2)
        X[i,:,0] = list(reversed([int(b) for b in format_str.format(nb1)]))
        X[i,:,1] = list(reversed([int(b) for b in format_str.format(nb2)]))
        T[i,:,0] = list(reversed([int(b) for b in format_str.format(nb1 -␣
 ↪nb2)]))
    return X, T
```

[3]:

```
sequence_len = 12
nb_train = 2000
X_train, T_train = create_sum_dataset(nb_train, sequence_len)
printSample(X_train[0,:,0], X_train[0,:,1], T_train[0,:,:])
```

```
x1: 101001000010   1061
x2: + 110101110000    235
      -------   ----
t:  = 000010001010   1296
```