

binary_input_generator.py

```
#!/usr/bin/env python3
import argparse

import numpy as np

# example call:
# python3 binary_input_generator.py data/random_0_10_16x16_fp32 16*16 random csv float32 -rand_lo 0.0 -rand_hi 10.0
parser = argparse.ArgumentParser()
parser.add_argument("filename")
parser.add_argument("N")
parser.add_argument("generator", choices=["const", "random", "incr"], default="random")
parser.add_argument("fileextension", choices=["csv", "bin", "dat"], default="dat")
parser.add_argument("datatype", choices=["float32", "float64"], default="float64")
parser.add_argument("--rand_lo")
parser.add_argument("--rand_hi")
parser.add_argument("--const_val")
args = parser.parse_args()

# general parameters
_N = int(eval(args.N))
_GENERATOR = args.generator # {const, random, incr}
_FILENAME = args.filename
_FILE_EXTENSION = args.fileextension # {csv, bin, dat}
_DATA_TYPE = np.dtype(args.datatype)
# random
_LOWER_BOUND = float(args.rand_lo) if args.rand_lo is not None else 0.0
_UPPER_BOUND = float(args.rand_hi) if args.rand_hi is not None else 0.0
# constant
_CONSTANT = float(args.const_val) if args.const_val is not None else 0.0

# generate data
if _GENERATOR == "random":
    # random data
    data = np.array(np.random.uniform(low=_LOWER_BOUND, high=_UPPER_BOUND, size=_N), dtype=_DATA_TYPE)
elif _GENERATOR == "const":
    # constant value
    data = np.array([_CONSTANT]*_N, dtype=_DATA_TYPE)
elif _GENERATOR == "incr":
    # increasing value
    data = np.arange(0.0, _N, dtype=_DATA_TYPE)
else:
    print("Generator {} not supported. Exit.".format(_GENERATOR))
    exit()

# write data to file
filename = _FILENAME + "." + _FILE_EXTENSION

if _FILE_EXTENSION == "csv":
    np.savetxt(filename, [data], delimiter=",")
```

`binary_input_generator.py`

```
elif _FILE_EXTENSION == "bin" or _FILE_EXTENSION == "dat":  
    data.astype(_DATA_TYPE).tofile(filename)
```