AGENDA

Interactive course: Reproducible Science and (Deep) Software (Variability)

Is (x+y)+z == x+(y+z)? (TP1)

Is (x+y)+z == x+(y+z)?

Is (x+y)+z == x+(y+z)?

In Informatics, not in Mathematics

x, y, z: number

Choose the language, the compiler/interpreter, the library, the computing environment, the computer you want…

And tell us!

Is (x+y)+z == x+(y+z)?

In Informatics, not in Mathematics

x, y, z: number

Choose the programming language (PL), the compiler/interpreter, the library, the computing environment, the computer you want…

Is your work reproducible?

PL = Java, Rust, Scratch, SageMath, etc.

~~Is (x+y)+z == x+(y+z)?~~

How often (x+y)+z == x+(y+z)?

~~Is (x+y)+z == x+(y+z)?~~

How often (x+y)+z == x+(y+z)?

write a program that returns a percentage

again: Choose the programming language, the compiler/interpreter, the library, the computing environment, the computer you want…

https://twitter.com/StasBekman/status/1749480373283905611

```
In [1]: import torch, struct
   ...: def binary_double(num):
   ...:     print(''.join(f'{c:0>8b}' for c in struct.pack('!d', num)))
   ...: binary_double(torch.tensor(9, device='cuda')/10)
   ...: binary_double(torch.tensor(9, device='cpu' )/10)
0011111111101100110011001100110011100000000000000000000000000000
0011111111101100110011001100110011000000000000000000000000000000
```

Stas Bekman @StasBekman · Jan 20

Floating point math discrepancies with some pretrained LM models can be an issue.

I was debugging today a weird discrepancy between Llama-2-7b inference results which proved to be triggered by whether `from_pretrained` was called ...
Show more

This is from the mps device:

Code | Blame | 58 lines (53 loc) · 1.2 KB

```c
#include <stdio.h>
#include <stdlib.h>
#define XMASK (char)(99)
static int randseed=-1;

/* Macro to return a uniform distributed random real number between 0 and 1 */
/* s is the seed, bb the multiplier. 'Maxint &' is used to mask the sign bit*/
#define Maxint (0x7fffffff)
#define invmaxint 4.656612875e-10
#define bb 31415821
#ifdef CUSTOM
#define uniform (double)(invmaxint*(randseed=(Maxint & ((randseed*bb)-1))))
#else
#ifdef WIN
#define uniform rand()
#else
#define uniform drand48()
#endif
#endif

int associativity_test()
{
  double x = uniform; double y = uniform; double z = uniform;
  //  printf("%f   %f   %f   %d\n",x,y,z,x+(y+z) == (x+y)+z);
  return x+(y+z) == (x+y)+z;
}

float proportion(int number)
{
  int ok=0;
  for (int i=0;i<number;i++) ok += associativity_test();
  return ok*100.0/number;
}


/* Here is the entry point of the program */
#ifdef OLD_MAIN_C
void main(argc, argv)
int argc;
char argv[];
{
#else
int main(int argc, char **argv) {
#endif
#ifdef WIN
  srand(123);
#else
  srand48(123);
```

https://github.com/FAMILIAR-project/reproducibility-associativity/

Language	Library	System	Compiler	VariabilityMisc	EqualityCheck	NumberGenerations	Repeat	min	max	std	mean
Perl				seed None	ASSOCIATIVITY	100	10	100.0	100.0	0.0	100.0
Perl				seed None	MULT_INV	100	10	60.0	71.0	3.562302626111375	65.1
Perl				seed None	MULT_INV_PI	100	10	51.0	63.0	3.330165161069343	55.9
Perl				seed 42	ASSOCIATIVITY	100	10	100.0	100.0	0.0	100.0
Perl				seed 42	MULT_INV	100	10	62.0	62.0	0.0	62.0
Perl				seed 42	MULT_INV_PI	100	10	47.0	47.0	0.0	47.0
Go				seed None	associativity	100	10	71.0	82.0	3.3466401061363023	76.0
Go				seed None	mult-inverse	100	10	58.0	78.0	6.0	66.0
Go				seed None	mult-inverse-pi	100	10	42.0	64.0	5.885575587824865	53.4
Go				seed 42	associativity	100	10	81.0	81.0	0.0	81.0
Go				seed 42	mult-inverse	100	10	70.0	70.0	0.0	70.0
Go				seed 42	mult-inverse-pi	100	10	56.0	56.0	0.0	56.0
R				seed None	ASSOCIATIVITY	100	10	100.0	100.0	0.0	100.0
R				seed None	MULT_INV	100	10	62.0	72.0	2.764054992217051	66.6
R				seed None	MULT_INV_PI	100	10	47.0	57.0	2.808914381037628	53.1
R				seed 42	ASSOCIATIVITY	100	10	100.0	100.0	0.0	100.0
R				seed 42	MULT_INV	100	10	67.0	67.0	0.0	67.0
R				seed 42	MULT_INV_PI	100	10	53.0	53.0	0.0	53.0
Julia				seed None strict-equality	ASSOCIATIVITY	100	10	74.0	90.0	4.6097722286464435	82.5
Julia				seed None strict-equality	MULT_INV	100	10	60.0	79.0	6.16765757804371	68.6
Julia				seed None strict-equality	MULT_INV_PI	100	10	49.0	59.0	2.8301943396169813	54.3
Julia				seed 42 strict-equality	ASSOCIATIVITY	100	10	89.0	89.0	0.0	89.0
Julia				seed 42 strict-equality	MULT_INV	100	10	73.0	73.0	0.0	73.0
Julia				seed 42 strict-equality	MULT_INV_PI	100	10	55.0	55.0	0.0	55.0
Julia				seed None approximate equality of Julia lang	ASSOCIATIVITY	100	10	100.0	100.0	0.0	100.0
Julia				seed None approximate equality of Julia lang	MULT_INV	100	10	100.0	100.0	0.0	100.0

https://github.com/FAMILIAR-project/reproducibility-associativity/

Please contribute!

new language
new compiler/interpreter
new options/flags

source code
Dockerfile, nix, guix

https://github.com/FAMILIAR-project/reproducibility-associativity/

~~Is (x+y)+z == x+(y+z)?~~ **~~How often (x+y)+z == x+(y+z)?~~**

My banker proposed this investment to me:

You give me e ≈ 2.71828...

The following year, I take 1 euro as a fee and multiply by 1.

The next year, I take 1 euro as a fee and multiply by 2.

The next year, I take 1 euro as a fee and multiply by 3.

...

After n years, I take 1 euro as a fee and multiply by n.

To retrieve my money, there is a 1 euro fee.

In 50 years, for my retirement, how much money will I have?

Write a program that answers this question

https://www.college-de-france.fr/fr/agenda/seminaire/semantiques-mecanisees-quand-la-machine-raisonne-sur-ses-langages/arithmetique-des-ordinateurs-et-sa-formalisation

Sylvie Boldo talk (the previous example was based on her talk, thanks!)

Reproduce everything (figures, tables, etc.) that has been said!
Replicate! (change programming language, compiler flags or versions, etc.)