[544] PyTorch and Numbers

Tyler Caraza-Harter

Outline

PyTorch Overview

Numeric Types

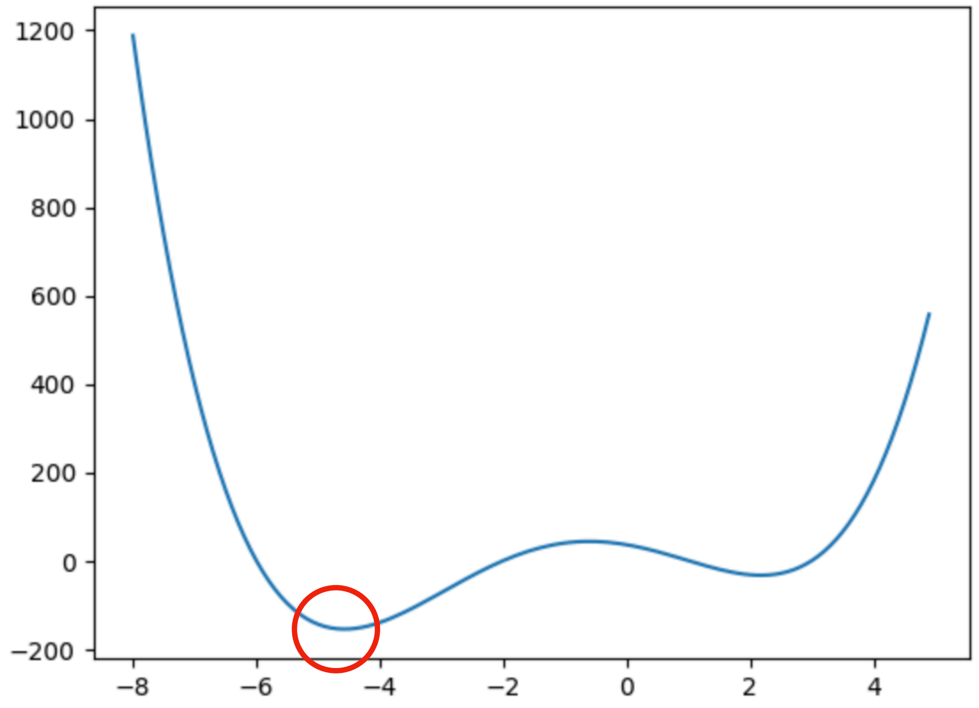
Coding Demos

- numeric types
- calculations: element wise, sigmoid, matrix multiplication, linear models
- optimization
- troubleshooting

PyTorch Uses

1. Floating point operations
   - scientific computing, machine learning
   - matrices, linear algebra
   - seamless: on CPU or GPU
   - distributed computing

2. Optimization
   - y = f(x)
   - which x makes y smallest? (or largest?)



3. Machine learning:
   - deep learning is an optimization problem
   - what parameters yield best performance metrics for some data?
   - sigmoid(sigmoid(data @ matrix1 + bias1) @ matrix2 + bias2)

Install

```
pip3 install -f https://download.pytorch.org/whl/torch_stable.html torch==1.13.1+cpu
pip3 install tensorboard
```

Outline

PyTorch Overview

Numeric Types

Coding Demos

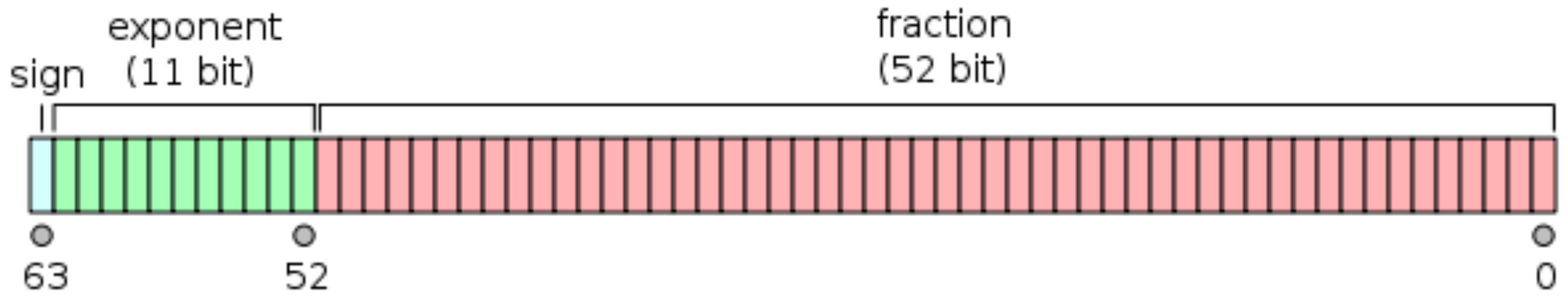
- numeric types
- calculations: element wise, sigmoid, matrix multiplication, linear models
- optimization
- troubleshooting

Python Numeric Types (Built In)

https://docs.python.org/3/library/stdtypes.html#numeric-types-int-float-complex

Python Types

- ints
  - ➡ no maximum/minimum size (Python is unusual in this way)
  - ➡ bigger/smaller values => more bits necessary
- floats
  - ➡ usually 64 bits ("double precision"; 32 bits would "single precision")
  - ➡ like exponential notation (1.23×10^2), but in binary instead of decimal
  - ➡ min/max size. Inf, -Inf, NaN have special bit combinations



https://en.wikipedia.org/wiki/Double-precision_floating-point_format

- complex
  - ➡ real and imaginary represented as two floats
  - ➡ not covered in 544

Other Numeric Types

Common numeric types that (a) CPUs can directly manipulate and (b) PyTorch supports

- integers: uint8, int8, int16, int32, int64
- floats: float16, float32, float64
- names specify bits, float vs. int, and signed ("u" => unsigned)
- dtype (data type)

```
import torch
x = torch.tensor(3.14, dtype=torch.float16)
```

PyTorch float16 ↑ (x)   Python float ↑ (3.14)

```
print(x.element_size())   # 2 bytes (instead of 8)
```

Tradeoffs: precision, range, memory usage

Outline

PyTorch Overview

Numeric Types

Coding Demos

- numeric types
- calculations: element wise, sigmoid, matrix multiplication, linear models
- optimization
- troubleshooting