Bitrounding

Decimal rounding?

		Sign	Exponent	Mantissa bits
0.123456789	≈ 0.1 =	0	01111011	11111001101011011101010

… hardly any simpler, so not easily compressible!

Solution: Round in binary instead!

Redundancies are very easy to compress!

Array

0.188482	≈	0.1875	=	0 01111100 10000000000000000000000
0.2034692	≈	0.203125	=	0 01111100 10100000000000000000000
0.35524988	≈	0.34375	=	0 01111101 01100000000000000000000
0.443963	≈	0.4375	=	0 01111101 11000000000000000000000
0.5630906	≈	0.5625	=	0 01111110 00100000000000000000000

Real information | 3 keepbits | Noise

Advantages

- **Removes noise/false information**
- **Can be combined with any lossless compressor**
- **Strong error bounds from floating-point numbers**
- **Fine-grained control on the error**

Usage

```
> from numcodecs import BitRound
> da = xarray.DataArray( ... )
> da.copy(data=BitRound(keepbits=7).encode(da.values))
```

Disadvantages

- **Depending on compressor, may under-exploit spatial correlations**
- **Removes insignificant gradients**

Bitrounding: Example