Example-rounding-single-precision

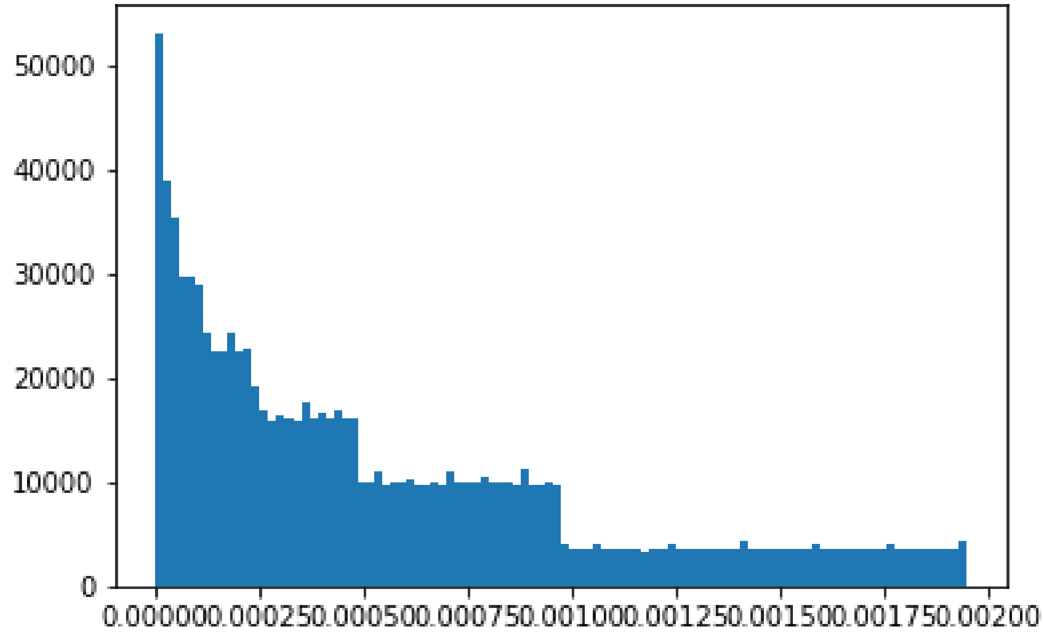
January 21, 2019

```
In [1]: import numpy as np

        def round_floats(vals, nbits=23):
            mask = 0xFFFF_FFFF << (23 - nbits)
            # Binary conversion and copy
            uvals = np.array( vals, dtype=np.float32 ).view(np.int32)
            # Apply mask
            uvals &= mask
            # Convert to single precision
            result = uvals.view(np.float32)
            return result
```

```
In [2]: # From 0. to 50.
        vals = np.array( 50*np.random.rand(1000000), dtype=np.float32 )
```

```
In [3]: import matplotlib.pyplot as plt
        %matplotlib inline

        # Example with 14 bits
        plt.hist( ( vals - round_floats( vals, nbits=14 ) ), bins=100 )
```

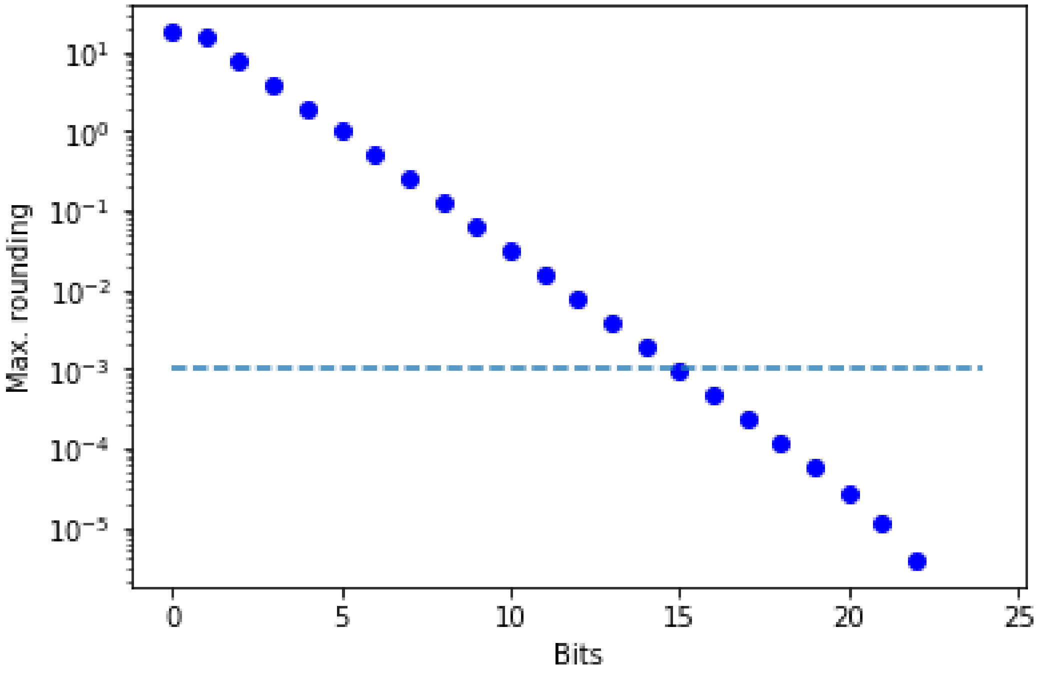
```
Out[3]: (array([53023., 38955., 35329., 29659., 29542., 28888., 24147., 22536.,
                22357., 24181., 22590., 22674., 19259., 16838., 15837., 16285.,
                16142., 15704., 17740., 16067., 16490., 15957., 16901., 16045.,
                15994., 10011.,  9842., 10938.,  9783.,  9848.,  9877., 10251.,
                 9730.,  9686.,  9851.,  9699., 10999.,  9817.,  9818.,  9817.,
                10454.,  9904.,  9839.,  9873.,  9625., 11091.,  9775.,  9758.,
                 9867.,  9765.,  4170.,  3569.,  3544.,  3482.,  4156.,  3609.,
                 3448.,  3613.,  3502.,  3627.,  3364.,  3493.,  3502.,  4112.,
                 3635.,  3541.,  3537.,  3436.,  3514.,  3505.,  3598.,  3649.,
                 4268.,  3551.,  3583.,  3457.,  3422.,  3574.,  3546.,  3548.,
                 3570.,  4128.,  3490.,  3542.,  3545.,  3552.,  3593.,  3474.,
                 3595.,  3458.,  4045.,  3586.,  3541.,  3506.,  3475.,  3563.,
                 3532.,  3524.,  3430.,  4238.]),
         array([0.0000000e+00, 1.9493104e-05, 3.8986207e-05, 5.8479309e-05,
                7.7972414e-05, 9.7465512e-05, 1.1695862e-04, 1.3645172e-04,
                1.5594483e-04, 1.7543793e-04, 1.9493102e-04, 2.1442413e-04,
                2.3391724e-04, 2.5341034e-04, 2.7290345e-04, 2.9239655e-04,
                3.1188966e-04, 3.3138276e-04, 3.5087587e-04, 3.7036894e-04,
                3.8986205e-04, 4.0935515e-04, 4.2884826e-04, 4.4834136e-04,
                4.6783447e-04, 4.8732758e-04, 5.0682068e-04, 5.2631379e-04,
                5.4580689e-04, 5.6530000e-04, 5.8479310e-04, 6.0428621e-04,
                6.2377931e-04, 6.4327242e-04, 6.6276552e-04, 6.8225863e-04,
                7.0175173e-04, 7.2124484e-04, 7.4073789e-04, 7.6023099e-04,
                7.7972410e-04, 7.9921720e-04, 8.1871031e-04, 8.3820341e-04,
                8.5769652e-04, 8.7718962e-04, 8.9668273e-04, 9.1617584e-04,
                9.3566894e-04, 9.5516205e-04, 9.7465515e-04, 9.9414820e-04,
                1.0136414e-03, 1.0331344e-03, 1.0526276e-03, 1.0721206e-03,
                1.0916138e-03, 1.1111068e-03, 1.1306000e-03, 1.1500930e-03,
                1.1695862e-03, 1.1890793e-03, 1.2085724e-03, 1.2280655e-03,
                1.2475586e-03, 1.2670517e-03, 1.2865448e-03, 1.3060379e-03,
                1.3255310e-03, 1.3450241e-03, 1.3645173e-03, 1.3840103e-03,
                1.4035035e-03, 1.4229965e-03, 1.4424897e-03, 1.4619827e-03,
                1.4814758e-03, 1.5009689e-03, 1.5204620e-03, 1.5399551e-03,
                1.5594482e-03, 1.5789414e-03, 1.5984344e-03, 1.6179276e-03,
                1.6374206e-03, 1.6569138e-03, 1.6764068e-03, 1.6959000e-03,
                1.7153930e-03, 1.7348862e-03, 1.7543792e-03, 1.7738724e-03,
                1.7933655e-03, 1.8128586e-03, 1.8323517e-03, 1.8518448e-03,
                1.8713379e-03, 1.8908310e-03, 1.9103241e-03, 1.9298173e-03,
                1.9493103e-03], dtype=float32),
         <a list of 100 Patch objects>)
```



```
In [4]: max_vals = []
        bits = []
        for i in range(24):
            bits.append( i )
            rounded = round_floats( vals, nbits=i )
            max_vals.append( np.max( vals - rounded ) )
```

```
In [5]: plt.plot(bits, max_vals, 'bo')
        plt.yscale('log')
        #plt.axis(ymin=0., ymax=1.)

        precision = 0.001

        xlin = np.linspace(0,24,100)
        plt.plot( xlin, [precision for i in range(len(xlin))], '--')

        plt.xlabel( 'Bits' )
        plt.ylabel( 'Max. rounding' )
```

```
Out[5]: Text(0, 0.5, 'Max. rounding')
```



```
In [6]: max_vals_arr = np.asarray( max_vals )
        sel = max_vals_arr < precision
        print ( sel )
        print ( max_vals_arr[ sel ] )
        print ( np.array( bits )[sel] )
```

```
[False False False False False False False False False False False False
 False False False  True  True  True  True  True  True  True  True  True]
[9.72747803e-04 4.84466553e-04 2.40325928e-04 1.18255615e-04
 5.72204590e-05 2.67028809e-05 1.14440918e-05 3.81469727e-06
 0.00000000e+00]
[15 16 17 18 19 20 21 22 23]
```