dm-test

November 5, 2018

1 Declarative memory test

```
In [1]: from dm import DM # import from dm.py in the same directory
        import numpy as np
        import matplotlib.pyplot as plt
```

Create a new DM:

```
In [2]: dm = DM()
```

Add some encounters of a few different chunks. They can be any type (e.g. number of pulses) but for clarity I'm using strings here. They are stored as key-value pairs in a dictionary. The key is the content of the chunk, the value is a list of encounters.

```
In [3]: dm.add_encounter(content = "banana", time = 0)
        dm.add_encounter("orange", 2)
        dm.add_encounter("pineapple", 6)
        dm.add_encounter("mango", 0)
        dm.add_encounter("mango", 1.5)
        dm.add_encounter("mango", 3.4)
        dm.add_encounter("mango", 6.99)
        dm.add_encounter("mango", 10.0)

        print(dm)
```

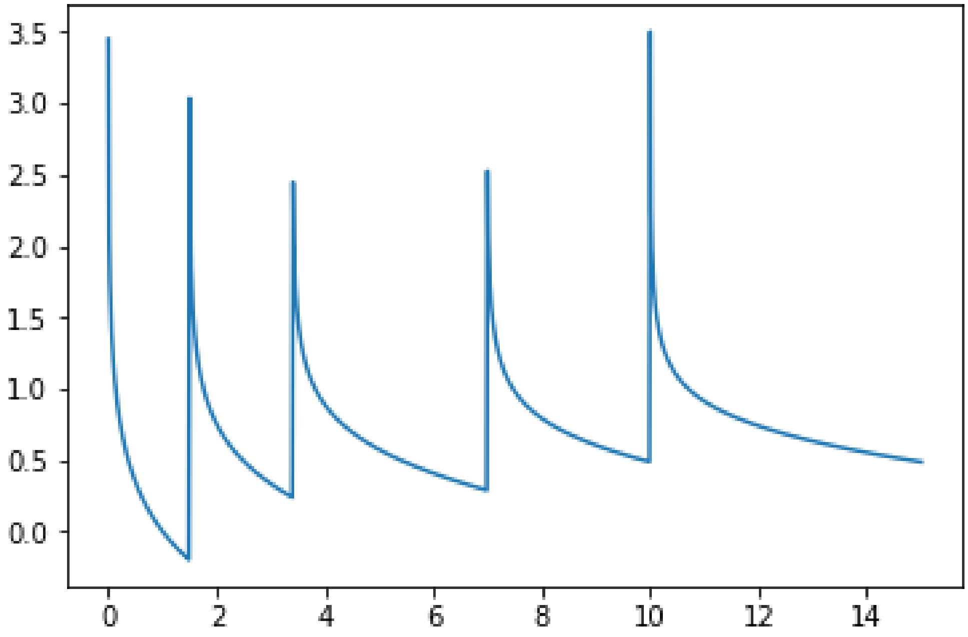
```
{'banana': [0], 'orange': [2], 'pineapple': [6], 'mango': [0, 1.5, 3.4, 6.99, 10.0]}
```

What is the baselevel activation of the "mango" chunk over time?

```
In [4]: x = np.linspace(start = 0, stop = 15, num = 1000)
        bl = []
        for i in x:
            bl.append(dm.get_baselevel_activation("mango", i + 0.001))

        plt.plot(x, bl)
```

```
Out[4]: [<matplotlib.lines.Line2D at 0x7f509fd709e8>]
```

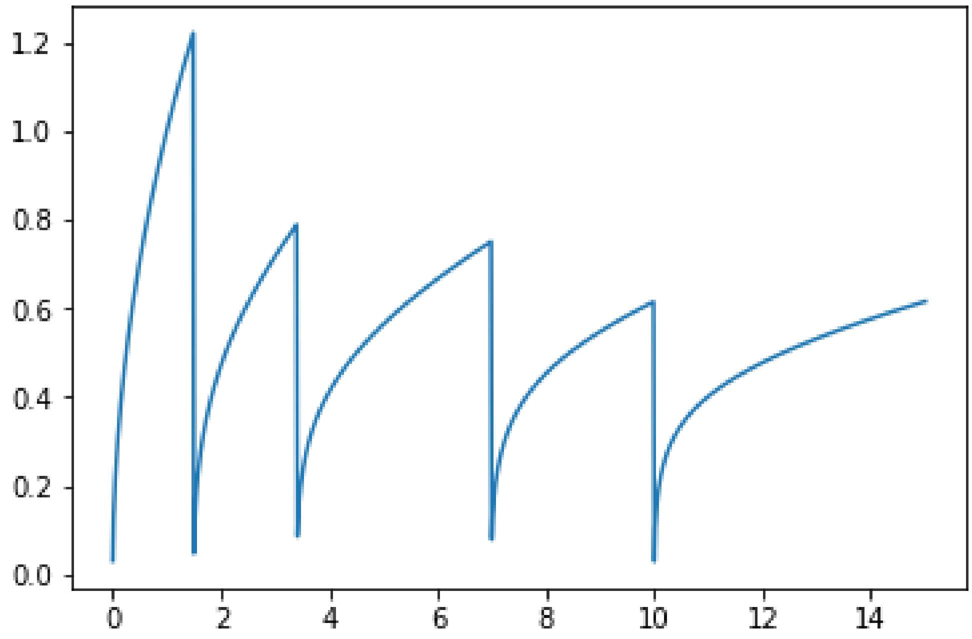


What is the retrieval latency of the "mango" chunk over time?

```
In [5]: x = np.linspace(0, 15, 1000)
        lat = []
        for i in x:
            lat.append(dm.get_latency("mango", i + 0.001))

        plt.plot(x, lat)

Out[5]: [<matplotlib.lines.Line2D at 0x7f509fca12b0>]
```


1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14