Sinus Generator

November 23, 2021

[4]:
```
import numpy as np
import matplotlib.pyplot as plt
import h5py as h5
import os
```

[5]:
```
def generate_sinus_data(n, hz_anim, add_channels_with_shift=0, add_noise=False):
    #hz_anim: a function that takes a single int as input and outputs a hz␣
 ↪number. (Must be defined for 0 to n)
    signal = np.empty((n, add_channels_with_shift+1))
    if add_channels_with_shift > 0:
        shift = np.random.randint(1, 30, add_channels_with_shift)
    for i in range(n):
        hz = hz_anim(i)
        signal[i, 0] = np.sin((i)/hz*2*np.pi) + add_noise*np.random.normal(0, 0.
 ↪03, 1)
        for j in range(add_channels_with_shift):
            signal[i, 1+j] = np.sin((i+shift[j])/hz*2*np.pi) + add_noise*np.
 ↪random.normal(0, 0.03, 1)
    print(signal.shape)
    signal = (signal-np.min(signal))/(np.max(signal)-np.min(signal))

    return signal

def save_h5(outpath, fname, data):
    with h5.File(os.path.join(outpath, fname), 'w') as f:
        f['data'] = data
        f.flush()
```

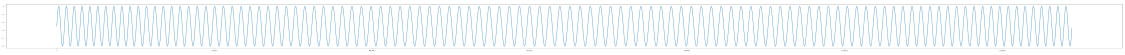
[73]:
```
signal = generate_sinus_data(115000, lambda x: (200/115000)*x+800,␣
 ↪add_channels_with_shift=11, add_noise=False)
#plt.plot(signal)
```

```
(115000, 13)
```

[ ]:
```
out = '/media/julian/Volume/data/sinus'
n_low, n_high = 115000, 150000
n_files = 100
hr_low , hr_high = 600, 1200
hr_fluc_low, hr_fluc_high = 100, 300
add_channels_with_shift=11
noise_chance = 0.1
for i in range(n_files):
    length = np.random.randint(n_low, n_high)
    noise = np.random.random() < noise_chance
    hr_fluc = np.random.randint(hr_fluc_low, hr_fluc_high)
    hr = np.random.randint(hr_low+hr_fluc, hr_high-hr_fluc)
    signal = generate_sinus_data(length, lambda x: hr+np.sin(x/
 ↪length*2)*hr_fluc, add_channels_with_shift=add_channels_with_shift,␣
 ↪add_noise=noise)
    save_h5(out, 'sinus'+str(i)+'.h5', signal)
```

[16]:
```
plt.figure(figsize=(100, 4))
plt.plot(signal[:, 0])
```

[16]: [<matplotlib.lines.Line2D at 0x7f602139a710>]



[ ]:
```

```