3.6

April 14, 2020

[ ]:
```
# %load 3.6
import numpy;
import matplotlib.pyplot as plt;

def func(w):
    a=abs(w)
    return a

def grad_g(w):
    if(w>=0):
        return 1
    else:
        return -1

alpha=0.5;
iterations=20;
w=numpy.zeros(iterations);
g=numpy.zeros(iterations);

ite=numpy.zeros(iterations);
w[0]=1.75;
g[0]=func(w[0]);

for i in range(iterations):
    ite[i]=i;

def fixed_alpha():
    for i in range(1,iterations):
      w[i]=w[i-1]-alpha*grad_g(w[i-1]);
      g[i]=func(w[i])
    plt.plot(ite, g)
    plt.show()


fixed_alpha()

def diminished_alpha():
```

```
    for i in range(1,iterations):
      w[i] = w[i - 1] - 1/i * grad_g(w[i - 1]);
      g[i] = func(w[i])
    plt.plot(ite, g)
    plt.show()

diminished_alpha()
```

[ ]: