Week5 Report

Yuchen.cai.uestc@gmail.com

1. Reproduce HWGQ using tf:

```
def get_hwgq(bitA):

    def quantize(x, k):
        # in order of
        assert k in [2,3,4,5], 'Does not support %d bits' % k
        code_book={                                  %写了一个编码表
                    %不同阶数对应的 level 不同
        '2':[0.5380, 0., 0.5380*(2**2-1)],
        '3':[0.3218, 0., 0.3218*(2**3-1)],
        '4':[0.1813, 0., 0.1813*(2**4-1)],
        '5':[0.1029, 0., 0.1029*(2**5-1)]
        }
        delta, minv, maxv = code_book[str(k)]
        #print(delta,minv,maxv)
        @tf.custom_gradient     %tf 自定义的一个梯度求导函数
        def _quantize(x):
            return
tf.to_float(x>0.)*(tf.clip_by_value((tf.floor(x/delta +
0.5)+tf.to_float(x<0.5*delta))*delta, minv, maxv)), lambda dy:
dy*tf.to_float(x>minv)*tf.to_float(x<maxv)

%前半句代码对应前向传播，后半句 lambda 对应反向传播
%优化过程可以存在于前向传播过程与反向传播过程

        return _quantize(x)

    def fa(x):
        if bitA == 32:
            return x
%如果输入的 bit 位数是 32，则直接返回输入 x，不进行 quantize 处理
        return quantize(x, bitA)
    return fa
% 没有理解这里的 return 为什么是 fa
```

2. Write HWGQ layer with CUDA

CUDA 编程不是很熟练，仍在学习中