

这周主要回顾了之前的计算卸载的论文，对主要的问题进行了梳理：

1：在基于二进制的计算卸载方案中，可以将问题归纳为一个非确定多项式问题 (NP-hard)。为了解决这个凸优化问题，一般都是用 KKT(KarushKuhn-Tucker) 条件求解得到最优值。然后通过博弈或者深度学习的方法进行训练。其中比较困难的在于凸优化问题的求解和边缘计算模型的建立。

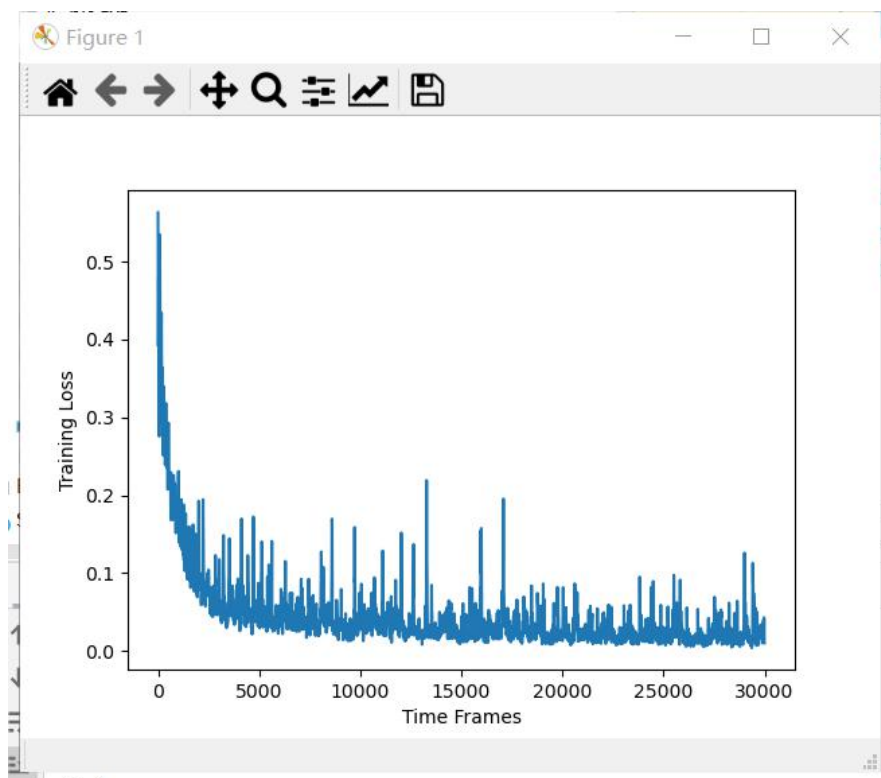
2：在基于部分任务卸载的计算卸载方案中，最关键的问题是如何卸载计算任务的多少，和分割程序。很多研究使用 TDMA/OFDMA 来设计控制卸载数据大小和时间/子信道的分配的最佳策略。

3：基于随机任务模型的计算卸载方案相关的文献还没有阅读，接下来也是学习的方向。

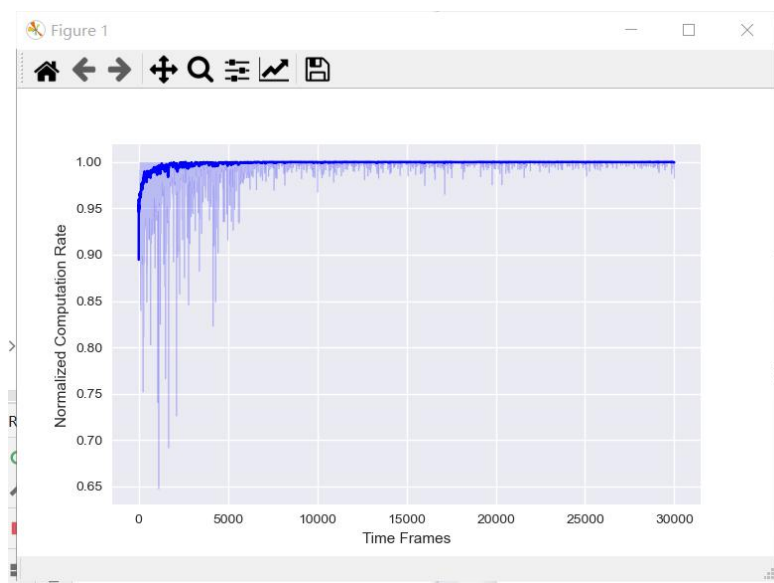
然后跑了 DROO 算法的实验代码，首先我们设置了一些基本参数，在 python 上使用 tensorflow，并设置了 DNN 层数为 4 其中第一个和第二个隐藏层分别有 120 个隐藏神经元，输入时间帧为 30000，记忆库尺寸为 1024，用户数量为 10。此外，将学习率(Learning rate)设置为 0.01，样本 尺寸(Batch Size)设置为 100。如图所示：

```
2.2.0
#user = 10, #channel=30000, K=10, decoder = 0P, Memory = 1024, Delta = 32
WARNING:tensorflow:From D:\anaconda\lib\site-packages\tensorflow\python\compat\v2_compat.py:96: disable_resou
Instructions for updating:
```

通过枚举所有个 2 的 N 次方个卸载动作得到最优解，可以得到学习率为 0.01 时的训练丢失率。如图所示：



并得到学习率为 0.01 时的奖励比率



在图中蓝色曲线表示奖励比率在过去 50 帧中的移动平均值，浅蓝色阴影代表奖励比率在过去 50 帧中的最大值和最小值，由图可见，随着时间的增加，奖励比率逐渐收敛到最优解。

其中关键的问题是数据样本如何获得，根据文献可知是使用 CD-method 来获得数据样本，下周工作主要阅读 “Computation rate maximization for

wireless powered mobile-edge computing with binary computation offloading," IEEE Trans. Wireless Commun., vol. 17, no. 6, pp. 4177-4190, Jun. 2018.来学习生成数据样本。