A Computational Complexity

Below, we are going to show the calculations that led to the results on the complexity of the algorithms present in the table of the complexity of algorithms. The analysis will be performed for the original approach OCDM, the Dataset-based OCDM and our method BAT-OCDM.
We are going to proof that our algorithm has a complexity wrt the number of tasks T that is logarithmic, while the complexity of OCDM and Datased-based OCDM is lineart wrt T.
While, in the original paper is showed only the complexity for batch, we insight further and analyze the complexity for the training of an entire task and the training on all tasks.
We are going to analyze as first, the original approach OCDM. In the following section, we are going to assume that each task has the same dataset size D and the memory $\mathcal{M}$ has a fixed size M.

A.1 OCDM

Studying the complexity for task, for each task we are going to iterate on the dataset $\mathcal{D}$ $\frac{D}{b}$ times, assuming that each batch will have size b.
In the original approach was proved that the complexity for the algorithm $Memory\_Update$(MU) using a memory of size M and removing b elements i.e. $MU(M, b)$ is:

$$MU(M, b) = O(b \cdot M - \frac{b \cdot (b-1)}{2}) \tag{6}$$

OCDM algorithm use Memory_Update to update the memory using a batch B of size b. In other words the algorithm MU given a set of M+b elements will remove b elements from it. The complexity of this is:

$$MU(M+b, b) \overset{(6)}{=} O(b \cdot (M+b) - \frac{b \cdot (b-1)}{2}) \tag{7}$$

to update a memory of size M removing b elements. The complexity of algorithm OCDM for a task consist in $\frac{D}{b}$ iterations over the MU method as indicated in (7), which correspond to:

$$\begin{aligned} OCDM_t(D, M) &= O(\frac{D}{b} \cdot [b \cdot (M+b) - \frac{b \cdot (b-1)}{2}]) \\ &= O(D \cdot [M + \frac{b+1}{2}]) \\ &\overset{M>>b}{=} O(D \cdot M) \end{aligned} \tag{8}$$

Where $OCDM_t(D, M)$ is the complexity of the algorithm OCDM for a single task, assuming the number of the samples of the task as D and the memory size used as M.

Since the entire training correspond to iterate on T tasks, we obtain as overall performance for OCDM:

$$OCDM(D, M) = \sum_{t \in \mathcal{T}} OCDM_t(D, M) = O(T \cdot D \cdot M) \tag{9}$$

A.2 Dataset-based OCDM

Though the update per batch is essential for the Online Continual Learning(OCL) setting, we are obtain suboptimal solutions respect to find the optimal distribution using all dataset D at once. Since we are evaluating the DIL scenario where the data D of a task is received all together we also study the performance of MU(M,D) which correspond to the of the variant *Dataset-based OCDM* (Db-OCDM), while the original will remain OCDM.

$$\begin{aligned} Db - OCDM(D, M)_t &= MU(D + M, D) \\ \overset{(6)}{=} O(D \cdot (M + D) - \frac{D \cdot (D-1)}{2}) &\overset{D >> 1}{=} O(M \cdot D + \frac{D^2}{2}) \end{aligned} \tag{10}$$

Since the entire training correspond to iterate on T tasks, we obtain as overall performance for Dataset-based:

$$Db - OCDM(D, M) = O(T \cdot [D \cdot M + \frac{D^2}{2}]) \tag{11}$$

A.3 BAT-OCDM

Below we show the complexity obtain considering our approach *BAT-OCDM*. In this case, the complexity is splitted in two subprocesses. The first one consist during the Task i to select $\frac{M}{i}$ samples from the data of the new task. Therefore the complexity of first part is equivalent to OCDM per task i.e. $OCDM_t(D, \frac{M}{i})$ will be $O(D \cdot \frac{M}{i})$.

$$OCDM_t(D, \frac{M}{i}) \overset{(8)}{=} O(D \cdot \frac{M}{i}) \tag{12}$$

The memory of an old task must be reduced from $\frac{M}{i-1}$ to $\frac{M}{i}$, therefore eliminating $\frac{M}{i \cdot (i-1)}$ samples.

To do this the complete complexity is $O(\frac{M^2}{i-1})$. In fact, we are going to perform $MU(\frac{M}{i-1}, \frac{M}{i \cdot (i-1)})$ i-1 times during task i(assuming tasks start from 1 to T included). Therefore, for the second part we have:

$$\begin{aligned}
(i-1)\cdot MU(\frac{M}{i-1},\frac{M}{i\cdot(i-1)}) &\overset{(6)}{=} \\
(i-1)\cdot O(\frac{M}{i\cdot(i-1)}\cdot\frac{M}{i-1}-\frac{M^2}{2\cdot i^2\cdot(i-1)^2}) &= \\
O(\frac{M^2}{i\cdot(i-1)}-\frac{M^2}{2i^2\cdot(i-1)}) &= \\
O(\frac{M^2}{i\cdot(i-1)}\cdot(1-\frac{1}{2\cdot(i)})) &= \\
O(\frac{M^2}{i\cdot(i-1)}) &= \\
O(\frac{M^2}{i-1})
\end{aligned} \tag{13}$$

Therefore, we have the total complexity for task i-th of algorithm BAT-OCDM is:

$$BAT-OCDM_t(D,M,i)=(13)+(12)=O(\frac{D\cdot M}{i}+\frac{M^2}{i-1}) \tag{14}$$

Where if i=1 then the second member is 0 since during i=1 there aren't old tasks to update. To evaluate the complexity of the entire training the calculations are trivial for OCDM and Dataset-based OCDM because it is enough to multiply the complexity by task for the value T, which is the total number of tasks.
In the case for BAT-OCDM is a little more tricky because the task complexity depends by task i.
To calculate is necessary to consider the following inequality:

$$\sum_{i=1}^{n}\frac{1}{i}\leq\ln n+1 \tag{15}$$

Therefore, for the first part:

$$O(\sum_{i=1}^{T}D\cdot M\cdot\frac{1}{i})\overset{(15)}{=}O(D\cdot M\cdot(\ln T+1)) \tag{16}$$

In the same way we have for the second part:

$$\begin{aligned}
O(\sum_{i=2}^{T}\frac{M^2}{i-1}) &\overset{(k=i-1)}{=} O(\sum_{k=1}^{T-1}\frac{M^2}{k}) \\
&\overset{(15)}{=} O(M^2\cdot[\ln(T-1)+1])=O(M^2\cdot[\ln(T)+1])
\end{aligned} \tag{17}$$

In total we have:

$$\begin{aligned}
BAT-OCDM(D,M) &= (16)+(17) \\
&= O((\ln T+1)\cdot(D\cdot M+M^2))
\end{aligned} \tag{18}$$