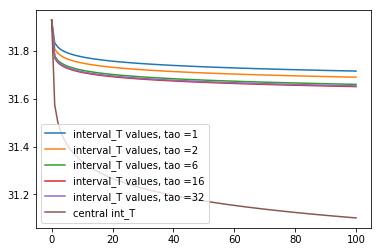
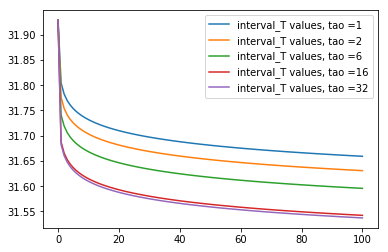
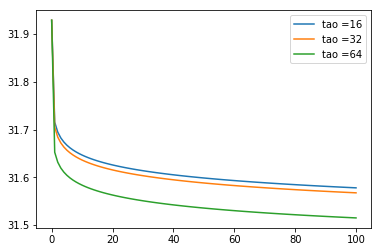
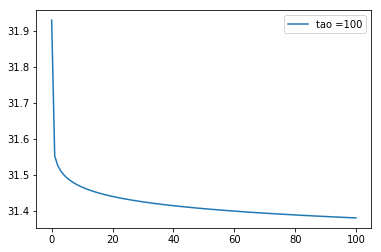
1. There is a convergence limit if each worker goes through its local dataset multiple times (tao). As tao increases, there is a significant gap from the centralized curve. This implies the necessity for communication between different nodes. The experiment has N =2 workers, without data exchange.



Zoom in:

1. 
2. 
3. 

**Some observations**

1. At least for the linear regression problem, if we da data shuffling after each global update, no matter what tao is , the convergence does not seem to be improve upon non-shuffling data, and gets even worse.
2. There should be a smarter way to shuffling, including when to shuffle and when to stop shuffling; the amount of information being exchange (shuffling parameter); the shuffling scheme;
3. The general interpretation: enough number of local iterations without contacting the master leads to the convergence to individual local minima. The average of the local minima will be the final convergence point of the federated learning algorithm if tao is large enough and one global update is performed at the ending point.
4. Under a given computation and communication budget, there is a way to schedule the size of tao along the iterations and global updates.
5. Shuffling after each global update may not help at all from the averaged local minima perspective. The reason is : each shuffling changes the local dataset of the workers, and thus changes the local minima. In each local epoch, the weight starts from a point and heads towards its local minima. Frequent shuffling leads to a zig-zag shape of the convergence path, with the final direction determined by the last shuffled data, implying that all the previous shuffling do not have any effect on the final weight.
6. If we somehow can store the weight at those previous turning points where data shuffling is performed, we can do an average of the local minima for each worker w.r.t shuffled local dataset, the average on the set of which should be closer to the global minima. More specifically, we want to design the local datasets (via shuffling) such that the average of their individual (local) minima is close enough (equal at best) to the global minima on the whole dataset.
7. Nick’s idea: at each shuffling stage, only shuffle the data of the nodes with outlier weights, in the sense that they can be pulled to the center.