Reproducing Resampling Paper Figure 1a

Iman Tabrizian

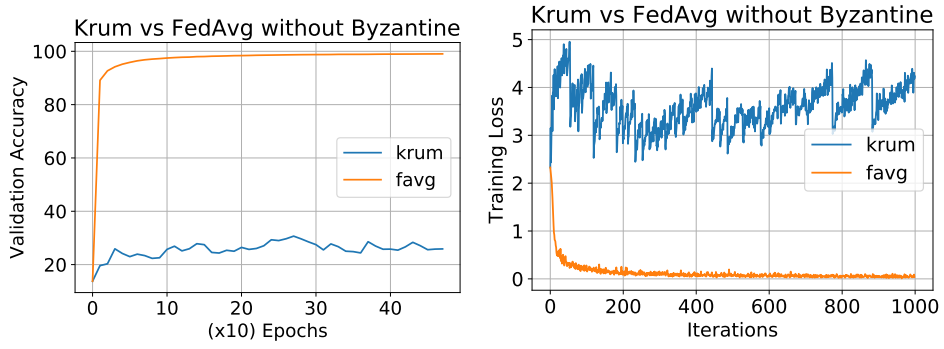
July 6, 2020

1 Results

In this section we present the results for training a CNN on MNIST. Fig. 1 shows that Krum[1] performs worse than FedAvg [2] when being trained on non-i.i.d. dataset. Fig. 3 shows that Krum favors certain clients more and some of the clients are almost never selected. The reason is that Krum has been designed for detecting byzantine workers in i.i.d. setting.

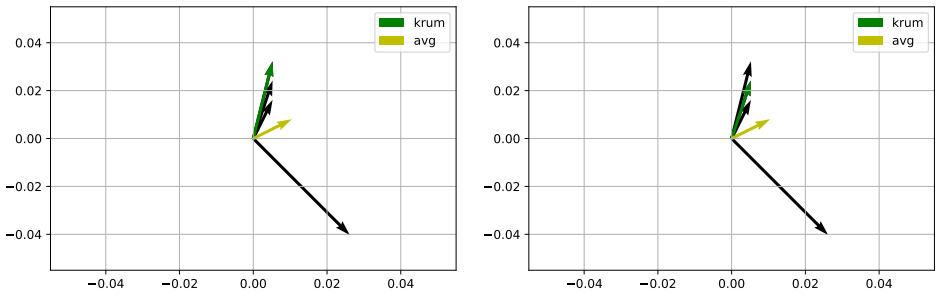
Fig. 2 shows the gradient selected by Krum when the number of byzantine workers is equal to 1 and 0. When the number of byzantine workers is 0 Krum chooses the gradient closest to the majority of gradients. In our very special case $n - f - 2 = 1$ (Fig. 2 left), Krum chooses the gradient with closest distance to another gradient. In this special case, the selected gradient is not unique. We only show one of the possible gradients.

It is important to note here that the optimization employed here does not use any momentum or weight decaying algorithm.



(a) Validation Accuracy **(b)** Training Loss

Figure 1: Training on Heterogeneous MNIST



(a) $f = 1$, the vector that is closest to one other vector is selected.

(b) $f = 0$, the vector that is closest to two other vectors is selected.

Figure 2: Krum Direction, green vector is the vector selected by Krum. Black vectors are some hypothetical vectors

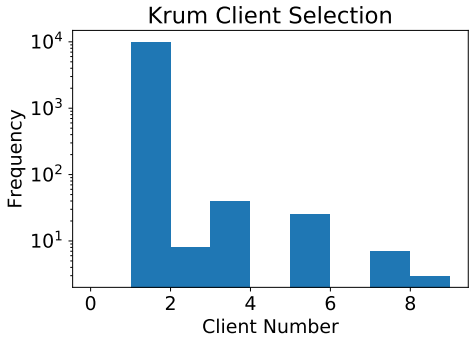


Figure 3: Krum Client Selection

References

Blanchard, P., Guerraoui, R., Stainer, J. et al. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in neural information processing systems* (pp. 119–129).

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282).