Addressing Statistical Heterogeneity through Generative Similarity-Based Comparison in Federated Learning

Aggregation Weight Modifications Using Latent Space Insights

Henry Page
h.page@student.tudelft.nl
Supervisors:
Dr. David Tax, Swier Garst

Introduction

Federated Learning sees a set of massively distributed clients train a global model in a collaborative fashion (Fig. 1) [1]. Local data on each client is secure and private, as data is kept on the clients. The distribution of the data however, is not IID, as certain clients may have biased data collection techniques. Category distribution heterogeneity (Fig. 2 and Fig.3) is a problem, as clients have different local objectives. Performance issues occur due to divergent optimisation directions [2].

Research Question

How does adapting aggregation weights in FL, based on the differences between the global latent space and locally encoded sample distributions of a VAE, affect model performance and convergence?

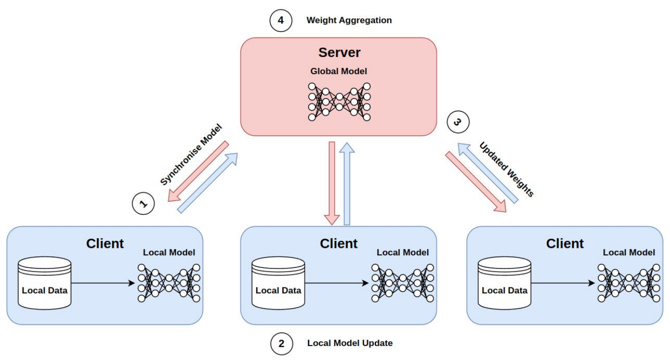

Fig 1: Overview of FL

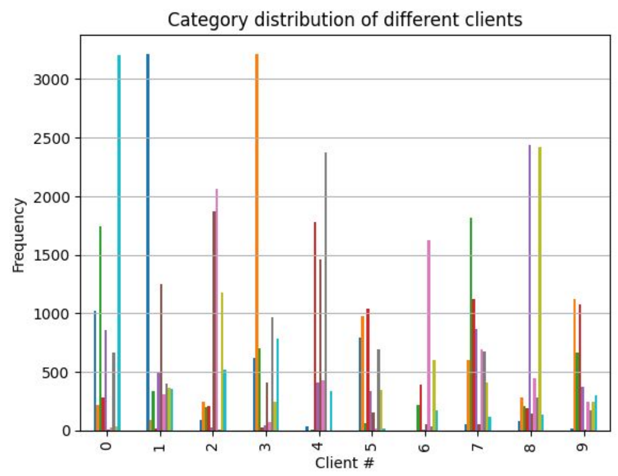

Fig 2: Visualisation of NIID-1 Class Distribution

Background: FedAvg & FedDisco

FedAvg uses relative dataset size to determine the weight of the client. Relative dataset size is not a good indicator of a client's importance by it self and clients should be weighed on their relative dataset size and local discrepancy instead. This algorithm is called *FedDisco* (Equation 1) [2].

$$p_k = \frac{(n_k - \alpha \cdot d_k + b)_+}{\sum_{m=1}^{K} (n_m - \alpha \cdot d_m + b)_+}$$

Equation 1: Determining Client Weight based off Discrepancy and Relative Dataset Size

Background: β-Variational Autoencoder (β-VAE)

- Generative Model
- Latent space representation of the input matches a pre-defined prior (usually standard normal) [3].
- Sum of reconstruction loss and KL-Loss used as the loss function
- Clients which have imbalanced datasets will not have local sample distributions which match the pre-defined prior N(0, I) (Fig. 4).

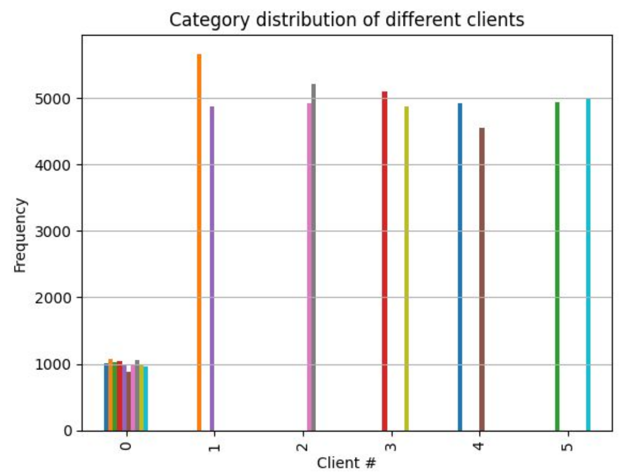

Fig 3: Visualisation of NIID-2 Class Distribution

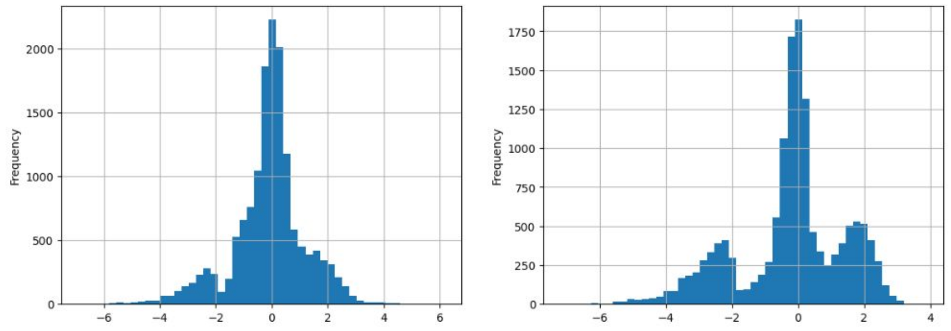

Fig 4: Local Encoding Distribution of IID Client and Non-IID Client.

function DETERMINE_NEW_WEIGHTS(K, Enc_ϕ, α, b)
  $w \leftarrow []$
  for each client $k \epsilon K$ **do**
    $\mathcal{E}^{m \times n} \leftarrow$ Generate embeddings of k's local samples through forward pass on Enc_ϕ
    $d_k \leftarrow 0$
    for each dim$\epsilon \mathcal{E}^\intercal$ **do** ▷ Each latent dimension
      $\mathcal{D} \leftarrow$ Wasserstein_Distance(dim, $\mathcal{N}(0,1)$)
      $d_k \leftarrow d_k + \frac{\mathcal{D}}{n}$
    end for
    $w[k] \leftarrow \mathrm{ReLU}(n_k - \alpha \cdot d_k + b)$ ▷ Equation 3
  end for
  $d_K \leftarrow \sum_{k=0}^{|K|-1} w[k]$
  for $k = 0, 1, \ldots |K| - 1$ **do**
    $w[k] \leftarrow \frac{w[k]}{d_K}$
    $p_k \leftarrow w[k]$
  end for
  return w
end function

Algorithm 1: Determining New Client Weights

Proposed Method (Algorithm 1)

- Encode local samples on each client.
- Compute the Wasserstein distance between the encoded values and a standard normal distribution
- Calculate the local discrepancy factor as the average of the distances over all latent dimensions.
- Use the local discrepancy factor for each client with equation 1 to calculate the new client weight.

Experimental Setup

- MNIST and FMNIST datasets.
- NIID-1 (Fig. 2) and NIID-2 (Fig. 3) with 10 and 6 clients respectively.
- 25 communication rounds, batch size of 64 and 10 local epochs.
- Different values of α (discrepancy coefficient) were tested.

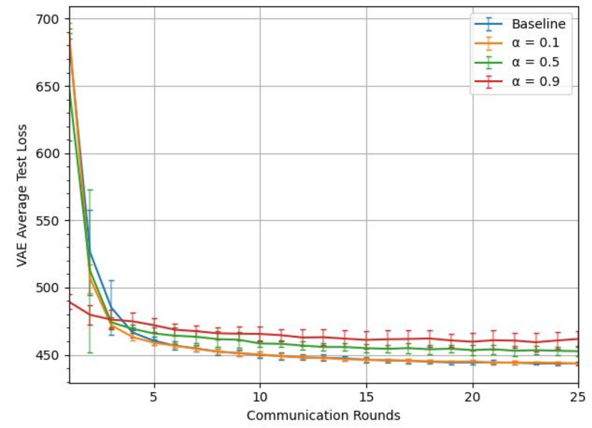

Fig 5: Performance in NIID-1 Scenario on FMNIST

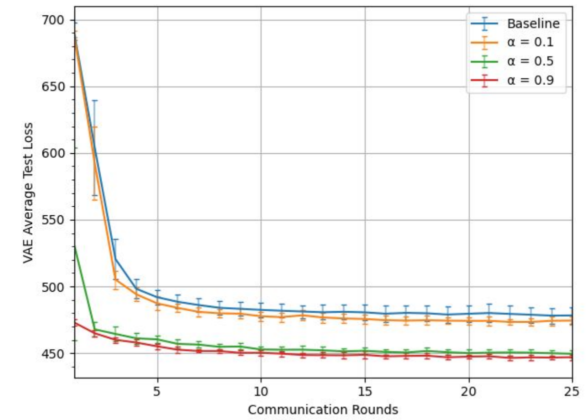

Fig 6: Performance in NIID-2 Scenario on FMNIST

Results & Conclusion

- NIID-1 (Fig. 5): led to initial performance increases. Overall effectiveness decreased as α increased. Results had a high degree of variation.
- In NIID-2 (Fig. 6), increasing α improved performance by up to 6.76%. Higher α values in NIID-2 reduced variance and enhanced performance, especially for FMNIST.
- Model performance and convergence were highly dependent on α. Higher α values in NIID-1 caused premature convergence, while α = 0.9 in NIID-2 showed performance gains.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data,"
[2] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "FedDisco: Federated learning with discrepancy-aware collaboration,"
[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes,"