Efficient Vertical Federated Learning: Leveraging Foundation Models to Improve Communication

Xavier Desprez and Oudom Kem

Universite Paris-Saclay, CEA, List, F-91120 Palaiseau, France
{xavier.desprez, oudom.kem}@cea.fr

Abstract. Vertical federated learning (VFL) is a variant of federated learning where participants share the same samples but with different features. We propose an approach that leverages a foundation model to extract features from data (images) for VFL training, which significantly reduces communication, while maintaining high accuracy. The approach was validated on a VFL benchmark dataset of satellite images, showing an improved accuracy of 1.5% and drastically reducing both the number of communication rounds (one-shot) and the communication size (by a factor 400).

Keywords: Vertical Federated Learning · Foundation Models · Computer Vision

1 Motivation

Vertical Federated Learning (VFL) is a variant of Federated Learning (FL) where parties share the same data instances but with different sets of features. VFL is particularly relevant in scenarios where different organizations hold private but complementary information about the same individuals. For example, a bank and an insurance company might each possess unique features about the same customers. Combining these features can lead to more accurate models.

Communication is an important bottleneck in Vanilla VFL as many communication rounds per iteration [1]. We focus on the one-shot communication approach, which aims to leverage it by extracting latent representations of data, traditionally using Self-Supervised Learning (SSL) and then communicating these representations only once to the label owner. To our knowledge, using foundation models for extracting these representations has not yet been explored, but we believe it could avoid computationally intensive SSL, improve performance, and retain the communication efficiency of the one-shot approach.

2 Method

2.1 Vanilla VFL training process

Parties and Data Distribution: Assume there are n parties $\{P_1, P_2, \ldots, P_n\}$ involved in the VFL process. Each party P_i holds a private dataset D_i consisting of the same set of samples but with different features. Formally, let

$D_i = \{(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)})\}$, where $x_i^{(j)}$ represents the j-th feature vector of the i-th party.

In VFL, one party (e.g., P_1) is designated as the label owner, also known as the "active party". This party aggregates intermediate representations from others and computes the final output, playing a key role in the process.

Training:

- Each party P_i computes intermediate representation $z_i^{(k)} = M_i(x_i^{(k)})$ with its local model M_i for each sample k and send them to the label owner P_1
- The label owner P_1 aggregates the received intermediate representations and computes the final output $y^{(k)} = M_G(z_1^{(k)}, z_2^{(k)}, \ldots, z_n^{(k)})$, then computes the gradients of the loss function with respect to the intermediate representations and sends these gradients back to the respective parties.
- Each party then uses these gradients to update their local model parameters.

These steps are repeated iteratively until the global model M_G converges to a desired level of accuracy. Each iteration requires communication between the parties and the label owner, making communication a critical limitation in VFL.

2.2 Proposed approach: VFL with a foundation model

We train our model using only one round of communication, leveraging foundation model for feature extraction (Dinov2 [2]).

- Each party P_i applies the foundation model to its local dataset D_i, generating feature vectors for each sample k
- The active party gets the combined feature dataset to train a centralized machine learning model $M_{central}$

3 Results

3.1 Dataset and experiment configuration

We utilized the Satellite dataset [3], an image dataset designed for VFL scenarios which comprises 16 revisits of Sentinel-2 satellites over approximately 3,000 Areas of Interest (AOIs). These 16 revisits are split among 16 parties, which aim to train jointly a model for classification problem with 4 classes.

We took the ResNet model from [3] as our baseline. For our model, we employed the frozen, pretrained Dinov2 model [2] with a linear head, leveraging Dinov's strength in achieving strong performance with a simple linear classifier. In this experiment, we utilized only the RGB channels from the 13 channels of Sentinel-2 images to fit with Dinov2 training data.

We used the following hyperparameters: a learning rate of 1×10^{-4}, the Adam optimizer with weight decay of 1×10^{-4}, StepLR scheduler with a step size of 10 and gamma of 0.5, 25 epochs, a batch size of 64, a 90%/10% train/validation split.

3.2 Experiment results

On random seeds 0 to 4 we had a mean result of 82.62% and a standard deviation of 0.54%, improving 1.5% the accuracy from [3] baseline (81.17%), and of 4% the accuracy of the best performing solo party with a communication budget reduced by a factor of 400, and only one communication versus 85,650.

Table 1. Comparison of Accuracy Results averaged over 5 seeds

Method	Data Used	Mean Accuracy	Standard Deviation	Communication Size (Mb)
Ours	VFL (P1-P15)	82.62%	0.54%	360.4
Baseline	VFL (P1-P15)	81.17%	0.35%	143982.1
Ours	Solo (Min - Max)	74.39% - 78.52%	0.32% - 0.27%	0
Baseline	Solo (Min - Max)	70.60% - 73.68%	1.19% - 0.86%	0

4 Conclusion

This study demonstrates that leveraging foundation models for feature extraction in Vertical Federated Learning (VFL) can significantly enhance communication efficiency and model accuracy. Our approach achieves a 1.5% improvement in accuracy while reducing communication size by a factor of 400 and limiting communication to a single round.

In the future, we aim to utilize non-RGB channels by converting them to grayscale images for feature extraction.We will also investigate security concerns regarding the potential invertibility of shared features, exploring options such as homomorphic encryption—viable due to the single linear layer—or incorporating an additional layer for passive parties.

References

1. Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.Q., Yang, Q.: Vertical Federated Learning: Concepts, Advances, and Challenges. IEEE Transactions on Knowledge and Data Engineering **36**(7), 3615–3634 (Jul 2024). https://doi.org/10.1109/TKDE.2024.3352628, https://ieeexplore.ieee.org/document/10415268/?arnumber=10415268, conference Name: IEEE Transactions on Knowledge and Data Engineering
2. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision (Feb 2024). https://doi.org/10.48550/arXiv.2304.07193, http://arxiv.org/abs/2304.07193, arXiv:2304.07193 [cs]
3. Wu, Z., Hou, J., He, B.: VertiBench: Advancing Feature Distribution Diversity in Vertical Federated Learning Benchmarks (Mar 2024), http://arxiv.org/abs/2307.02040, arXiv:2307.02040 [cs]