In the realm of machine learning, Federated Learning (FL) has emerged as a paradigm that allows for decentralized training across multiple devices or nodes. This approach ensures that raw data remains localized, addressing concerns related to data privacy and security. The primary objective of FL is to train a global model using data from various clients without the need to centralize this data. This decentralized approach is particularly beneficial in scenarios where data cannot be easily centralized due to privacy constraints or sheer volume.

Motivation:

In the realm of Federated Learning (FL), a significant challenge arises when the data across clients is heterogeneous. This is a realistic setting since data owned by clients participating in federated learning likely originates from different distributions. In such scenarios, the FL procedure might converge slowly, and the resulting global model may not perform well on an individual client's local data. To address this challenge, numerous FL methods have been proposed in recent years, aiming to enable learning on non-IID data. Unfortunately, these methods often struggle to train a robust global model when clients' data distributions vary significantly. Moreover, existing personalized FL methods typically improve the performance of local models at the expense of the accuracy of the global model.

Given these challenges, this paper introduces a novel approach, FedHKD (Federated Hyper-Knowledge Distillation), aiming to overcome the limitations.

Certainly! Let's focus on the Methodology section of the paper.

PROBLEM FORMULATION

In the framework presented in this paper, the authors delve into a federated learning system comprising m clients, each possessing their unique local private datasets, ranging from D1 to Dm. It's crucial to note that these datasets' distributions can differ significantly among clients. In some instances, a local dataset might only encompass samples from a subset of the total classes. Within this federated learning structure, there's a two-way communication: clients transmit their locally trained models to a central server, which, in turn, dispatches the consolidated global model back to the clients.

The foundational federated learning approach, as articulated by McMahan et al. (2017), employs the following aggregation formula: $wt=M1\sum i=1m \mid Di \mid wt-1i$ Where:

- wt symbolizes the parameters of the global model during the t-th round.

- $wt-1i$ represents the parameters of the i-th client's local model during the (t-1)-th round.
- m stands for the total number of clients in the system.
- M is the cumulative sum of all datasets, given by $M=\sum_{i=1m} \mid Di \mid$.

A common assumption in this setup is that all clients utilize a consistent model architecture. The overarching objective is twofold: firstly, to tailor a model for each client that excels on its specific dataset, and secondly, to ensure that each of these personalized models also augments the efficacy of the global model.

UTILIZING HYPER-KNOWLEDGE:

Knowledge distillation (KD) has gained prominence in federated learning, especially when anchored to a public dataset. Typically, clients use their local models to make predictions or inferences based on the samples from this dataset.

However, this methodology presents several challenges:

1. **Public Dataset Dependency**: Over-reliance on a public dataset can introduce biases. Such datasets might not always reflect the varied real-world scenarios the model will encounter.

2. **Computational Overhead**: Leveraging local models to make inferences on public datasets can be resource-intensive, more so if the dataset is extensive.
3. **Communication Overhead**: Transmitting and storing public datasets locally can demand significant communication bandwidth and memory.

To navigate these challenges, this paper introduces the innovative concept of "hyper-knowledge." Instead of solely relying on distilled knowledge, the authors expand this idea to encompass both the averaged data representations and their corresponding averaged soft predictions. This "hyper-knowledge" undergoes protection via the Gaussian differential privacy mechanism and facilitates sharing between the server and clients.

Mathematical Representation: For any given data sample, the model's output, denoted as z, can be transformed into soft predictions, q, using the equation: $q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$ Where:

- z_i represents the ith element of the model's output z.
- q_i is the ith element in the soft prediction q.
- T is the "temperature" parameter, influencing the softness of the predictions.

The "hyper-knowledge" encapsulates both the mean of these soft predictions and the mean of the data representations, offering a more comprehensive

context for the model. This enriched knowledge base paves the way for enhanced personalization and global model performance.

Feature Extractor and Classifier:

Image classification, the primary use case highlighted in this paper, offers a clear understanding of the role of deep networks. Such networks predominantly consist of two integral components:

1. **Feature Extractor**: This component is responsible for converting raw input data, like images, into a latent space representation. It essentially captures the underlying features or patterns in the data.
2. **Classifier**: Once the data is represented in the latent space, the classifier steps in to map these representations to categorical vectors, essentially assigning a label or category to the input data.

To put this into a mathematical context:

$h_i = R_{\phi_i}(x_i)$

$z_i = G_{\omega_i}(h_i)$ Where:

- x_i stands for the raw data corresponding to client i.
- $R_{\phi_i}(\cdot)$ and $G_{\omega_i}(\cdot)$ denote the embedding functions of the feature extractor and classifier, equipped with model parameters ϕ_i and ω_i respectively.

- h_i is the representation vector derived from x_i.
- z_i represents the categorical vector.

In layman's terms, the feature extractor processes an image, extracting crucial features or representations in a latent space. These extracted features encapsulate the core attributes of the image. Subsequently, the classifier interprets these features, determining the content or subject of the image.

Evaluating and Using Hyper-Knowledge:

In the process of understanding and utilizing hyper-knowledge, the computation of the mean latent representation for a specific class within a client's dataset is pivotal. For class $\boxed{?}j$ in the dataset of client $\boxed{?}i$, the mean latent representation is calculated as:

$$\bar{h}_{ji}=N_{ji}1\sum_{k=1}^{N_{ji}}h_{j,ki}$$

$$\bar{q}_{ji}=N_{ji}1\sum_{k=1}^{N_{ji}}Q(z_{j,ki},T)$$

Where:

- N_{ji} represents the count of samples labeled as j within the dataset of client i.
- $Q(\cdot,T)$ is the function that generates soft targets.
- $h_{j,ki}$ and $z_{j,ki}$ respectively denote the data representation and prediction for the hk_{th} sample labeled as j from client i.

The hyper-knowledge for class j in client i is essentially the combination of the mean latent data representation $\bar{h}_{ji}$ and the soft prediction $\bar{q}_{ji}$. For ease of reference, this is symbolized as $K_{ji}=(\bar{h}_{ji},\bar{q}_{ji})$. If the total number of classes is n, then the complete hyper-knowledge for client i is represented as $K_i=\{K_{1i},...,K_{ni}\}$.

Differential Privacy Mechanism

While communicating averaged data representation can promote privacy, the hyper-knowledge exchanged between the server and clients might still be susceptible to differential attacks. The scheme presented in this paper enhances privacy by safeguarding the shared hyper-knowledge using a Gaussian differential privacy mechanism.

GLOBAL HYPER-KNOWLEDGE AGGREGATION

This section is crucial as it elucidates how knowledge from various clients is amalgamated to form a comprehensive understanding.

To begin with, let's consider the equation where$pi=NjNij$. Here, Nij represents the number of samples in class j owned by client i, and Nj is the summation of Nij over all clients. This equation essentially gives us a weight or proportion of samples for each client in relation to a specific class.

Now, it's imperative to understand that $h\sim ij,t$ denotes the local hyper-knowledge about class j of client i at global round t. The noise in this equation, which is drawn from a normal distribution with mean 0 and variance $(Sfi)2\times\sigma 2$, has its impact on the quality of hyper-knowledge mitigated during aggregation. This is primarily because when we have a sufficiently large number of participating clients, the noise tends to average out.

Moving on to the next equation, which deals with the variance $\sigma 2\times\sum i=1m(Sfi)2$. This equation reinforces our earlier point: the additive noise, despite its presence, is effectively near-eliminated after the aggregation of local hyper-knowledge. For the sake of simplicity and to avoid any ambiguities, we assume that Nij is not equal to zero in the above expressions.

In essence, the Global Hyper-Knowledge Aggregation is a sophisticated mechanism that ensures the collective intelligence of all clients is harnessed, while simultaneously minimizing the noise. This ensures that the aggregated knowledge is both comprehensive and accurate.

Local Training Objective

The local training objective is crucial in federated learning as it determines how each client will train its model based on its local data.

The objective function is designed to ensure that the local model is aligned with the global model while also benefiting from the hyper-knowledge.

The equation is: $L(x,y,K,\lambda,\gamma)=CELoss(\omega i,\phi i(x),y)+\lambda\sum j=1n // Q(\omega i,\phi i(xj),T)-Kj // 22+\gamma\sum k=1Bi // \phi i(xk)-Hyk,t // 22$

Explanation:

1. **CELoss(\omega_{i}, \phi_{i}(x), y)**:
    - This term represents the **Cross-Entropy Loss**. It measures the difference between the predicted probabilities and the actual outcomes.
    - ωi is the local classifier for client i.
    - $\phi i(x)$ is the feature extractor for client i, which extracts features from the input data x.
    - y is the true label.
2. **\lambda \sum_{j=1}^{n} \lVert Q(\omega_{i}, \phi_{i}(x_{j}), T) - K_{j} \rVert_{2}^{2}**:
    - This term is a regularization term that ensures the soft target predictions are close to the global hyper-knowledge.
    - $Q(\omega i,\phi i(xj),T)$ represents the soft target function with temperature T. Soft targets are the class probabilities from the model's output when it's applied to the input data.

- Kj is the jth element of the global hyper-knowledge.
- λ is a hyper-parameter that controls the importance of this term in the loss function.

3. **\gamma \sum_{k=1}^{B_{i}} \lVert \phi_{i}(x_{k}) - H_{y_{k},t} \rVert_{2}^{2}**:

    - This term is another regularization term that ensures the extracted features are close to the global data representation.
    - $\phi i(xk)$ is the feature extracted from the kth data sample by client i's feature extractor.
    - Hyk,t is the global data representation for label ��yk at round t.
    - γ is a hyper-parameter that controls the importance of this term in the loss function.

In essence, the local training objective is designed to achieve a balance between making accurate predictions on the local data (first term) and aligning with the global knowledge (second and third terms). The hyper-parameters λ and γ control the trade-off between these objectives.

FedHKD (Federated Hyper-Knowledge Distillation) is a novel federated learning framework that leverages prototype learning and knowledge distillation to train on heterogeneous data.

Framework Overview:

The server initializes the global model, which consists of a feature extractor and a classifier.

At the start of each global epoch, the server sends the global model and global hyper-knowledge to selected clients.

Each client initializes its local model with the received global model and updates it by minimizing the objective function. This objective includes prediction loss, classifier loss (based on the distance between classifier output and global soft predictions), and feature loss (based on the distance between local and global data representations).

After local updates, clients compute their local hyper-knowledge and send both the local model and hyper-knowledge to the server for aggregation.

Explanation:

FedHKD is designed to address the challenges of data heterogeneity in federated learning. By combining prototype learning with knowledge distillation, it allows clients to compute mean representations and soft predictions for their local data classes. This "hyper-knowledge" is then aggregated at the server, ensuring that the global model benefits from the collective knowledge of all clients. The framework has been shown to consistently outperform other methods in terms of both local and global

accuracy.

Experiment and Results

The experiments and results section provides a comprehensive evaluation of the proposed method, FedHKD, in comparison with other state-of-the-art methods. The evaluation metrics include both local and global accuracy, and the experiments are conducted on various datasets and settings.

Key Highlights:

Datasets and Settings: The experiments are conducted on datasets like SVHN, CIFAR10, and CIFAR100. The number of clients varies, including experiments with 10, 20, and 50 clients. The data partitions are generated from the Dirichlet distribution with different concentration parameters.

Accuracy Comparison:

On the SVHN dataset, FedHKD significantly improves the local test accuracy over FedAvg by 19.5%, 14.3%, and 20.6% for experiments involving 10, 20, and 50 clients, respectively. The global test accuracy

also sees improvements of 37.0%, 15.6%, and 39.5% for the same client settings.

For the CIFAR10 dataset, the improvements over FedAvg are 5.1%, 8.9%, and 14.5% in local accuracy and 14.5%, 9.9%, and 45.6% in global accuracy for experiments with 10, 20, and 50 clients, respectively.

Table Insights:

Tables 1 and 2 present detailed results on data partitions generated from the Dirichlet distribution. The tables compare various schemes in terms of local and global accuracy, the number of parameters, and the time taken for each round of training.

The proposed method, FedHKD, generally ranks as either the best or the second-best in terms of both local and global accuracy. It competes closely with FedMD without using public data.

Additional Observations:

On CIFAR100, the improvement in global accuracy is somewhat more modest, but the improvement in local accuracy remains significant.

The local test accuracies of FedHKD* and FedProto are comparable, but FedHKD* outperforms FedProto in terms of global test accuracy.

In this research, the authors have presented a novel approach to federated

learning, introducing the Federated Hyper-Knowledge Distillation (FedHKD) framework. This methodology addresses the challenges of existing personalized FL methods, which often compromise the accuracy of global models to enhance local model performance.