Bayesian Federated Neural Matching That Completes Full Information

Peng Xiao[1], **Samuel Cheng**[2*]
[1]Department of Computer Science and Technology, Tongji University
Shanghai, China
[2]School of Electrical and Computer Engineering, University of Oklahoma
Oklahoma City, US
phd.xiaopeng@gmail.com

Abstract

Federated learning is a contemporary machine learning paradigm where locally trained models are distilled into a global model. Due to the intrinsic permutation invariance of neural networks, Probabilistic Federated Neural Matching (PFNM) employs a Bayesian nonparametric framework in the generation process of local neurons, and then creates a linear sum assignment formulation in each alternative optimization iteration. But according to our theoretical analysis, the optimization iteration in PFNM omits global information from existing. In this study, we propose a novel approach that overcomes this flaw by introducing a Kullback-Leibler divergence penalty at each iteration. The effectiveness of our approach is demonstrated by experiments on both image classification and semantic segmentation tasks.

Introduction

The recent decade has seen a rapid advancement in artificial intelligence, particularly in the subfield of deep learning. This is large because there is an abundance of data available. However, the increasing privacy concerns present a barrier to some datasets' accessibility in a number of applications, particularly in the medical and financial areas. Therefore, federated learning (FL), which involves learning a model from disparate sets of data, is suggested to address this issue.

Federated learning is a learning paradigm where locally learned models are combined into a shared global model. Neural networks are naturally used in federated learning since they play a vital role in deep learning. Considering the inherent permutation invariance of a neural network, it is reasonable to match neurons of distinct local models before aggregating them. To do this, Probabilistic Federated Neural Matching (PFNM) (Yurochkin et al. 2019b) builds a Bayesian nonparametric framework to match and merge these neurons. In the process of modeling, PFNM formally characterizes the generative process of local neurons through the Beta-Bernoulli process (BBP) (Thibaux and Jordan 2007), and treats the local neurons as noisy realizations of latent global neurons. In the process of optimizing, PFNM iteratively maximizes the posterior estimation of latent global neurons, through solving a linear sum assignment formulation of global neurons and current local neurons. PFNM subsequently extends in modern architectures such as convolutional neural networks (CNNs) and long short-term memory (LSTMs) (Wang et al. 2020), and its variants are also utilized in aggregating various statistical models such as Gaussian topic models, hierarchical Dirichlet process based hidden Markov models (Yurochkin et al. 2019a, 2018) etc.

In this paper, we theoretically prove the drawback of the optimizing process in PFNM under its probabilistic framework and address it by introducing a Kullback-Leibler (KL) divergence penalty. Our theory can also be generalized in all PFNM variants. Specifically, the contributions in this work include: 1) We theoretically prove that the linear sum formulation in the optimizing process omits the information of global neurons under the Bayesian framework. 2) We fix the missing global information by introducing a KL divergence penalty to complete the full information. 3) In experiment, we not only demonstrate the effectiveness of our approach on three image datasets, but also extend the matching type federated learning algorithm to neural network with batch normalization layer.

Related Works

Federated Learning The initial aggregation method in FL is FedAvg (McMahan et al. 2017) in which parameters of local models are averaged coordinate-wisely. But the performance of FedAvg deteriorates significantly in non-i.i.d (Independent and Identically Distributed) data (Karimireddy et al. 2020; Deng, Kamani, and Mahdavi 2021; Xiao et al. 2020). Subsequent improved methods starts from different perspectives. FedProx (Li et al. 2018) adds a proximal term in local training cost to keep dissimilarity between local models in lower bound. SCAFFOLD (Karimireddy et al. 2020) uses control variates to correct the drift in local updates. FedPD (Zhang et al. 2020) proposes a primal-dual optimization strategy to alleviate deterioration in non-i.i.d data. Agnostic federated learning (Mohri, Sivek, and Suresh 2019) optimizes a centralized distribution by maximize the worst local model. Federated multi-task learning (Smith et al. 2017) applies a multi-task learning mechanism to customize local models. Several studies further extend *knowledge distillation* (Hinton, Vinyals, and Dean 2015; Buciluǎ, Caruana, and Niculescu-Mizil 2006; Schmidhuber 1992) in

*is the corresponding author.

federated learning. And the key idea is to employ the knowledge of pre-trained teacher neural networks (local models) to learn a student neural network (global model) (Lin et al. 2020; Chen and Chao 2020; Zhu, Hong, and Zhou 2021). However, all those methods above don't considering the permutation invariance in neural network.

Parameter Matching There are also other researches about match the parameters. In (Singh and Jaggi 2020), the authors align neurons across different NNs by minimizing an optimal transportation cost matrix. However, it fixes the number of global neurons, making it impractical when data from different local models is extremely heterogeneous. Some work (Claici et al. 2020) uses variational inference to optimize the assignments between global and local components under a KL divergence. However, the optimization process is complex, and calculating variational inference is difficult.

Problem Formulation

Given S fully connected (FC) Neural Networks (NNs) with a hidden layer trained through different datasets: $f_s(\boldsymbol{x}) = \boldsymbol{W}_s^{(1)}\sigma(\boldsymbol{W}_s^{(0)}\boldsymbol{x})$, for $s = 1, \cdots, S$ (for simplifying notation, biases are omitted to), where $\sigma(\cdot)$ is the nonlinear activation function, $\boldsymbol{W}_s^{(1)} \in \mathbb{R}^{K\times J_s}$ and $\boldsymbol{W}_s^{(0)} \in \mathbb{R}^{J_s\times D}$ are the weights; with D being the input dimension, K being the output dimension (i.e., number of classes), and J_s being the number of neurons on the hidden layer of s-th NN. Neuron indexed by j is viewed as a concatenated vector $\boldsymbol{w}_{sj}^T = [\boldsymbol{W}_{s,j\cdot}^{(0)}, \boldsymbol{W}_{s,\cdot j}^{(1)^T}]$, where $j\cdot$ and $\cdot j$ denote the jth row and column correspondingly. And s-th FCNN can also be viewed as a collection of neurons $\{\boldsymbol{w}_{sj} \in \mathbb{R}^{D+K}\}_{j=1}^{J_s}$. In federated learning, we want to learn a global neural network with weights $\Theta^{(0)} \in \mathbb{R}^{J\times D}, \Theta^{(1)} \in \mathbb{R}^{K\times J}$, where $J \ll \sum_{s=1}^{S} J_s$ is an inferred variable denoting the number of global neurons.

Permutation Invariance Expanding the preceding expression of a FCNN: $f_s(\boldsymbol{x}) = \sum_{j=1}^{J_s}\sigma(\langle \boldsymbol{W}_{s,j\cdot}^{(0)}, \boldsymbol{x}^T\rangle)\boldsymbol{W}_{s,j\cdot}^{(1)}$. Summation is a permutation invariant operation, thus any permutation $\tau(1, \cdots, Js)$ of rows of $\boldsymbol{W}_s^{(0)}$ and columns of $\boldsymbol{W}_s^{(1)}$, i.e. the neurons, will not affect the output for any input $\boldsymbol{x}$. Due to the permutation invariance, a neuron indexed by j from one FCNN is unlikely to correspond to a neuron with the same index from another FCNN. Thus, we should match neurons from different FCNNs before aggregate them into a collection of global neurons $\{\boldsymbol{\theta}_i \in \mathbb{R}^{D+K}\}_{i=1}^{J}$.

Probabilistic Modeling

PFNM models the generative process of observed local neurons using a Beta Bernoulli process, which is described in the Appendix A, to infer the global model while accounting for the inherent permutation invariance. First, consider the collection of global neurons as prior which are sampled from a Beta process with a base measure: $M = \sum_i m_i\delta_{\boldsymbol{\theta}_i} \sim \text{BP}(1, \gamma_0 H)$ and $\boldsymbol{\theta}_i \sim H$, where γ_0 is the mass parameter, m_i are the stick-breaking weights, H is the base measure and chosen as a multivariate Gaussian distribution $H = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma_0})$ with $\boldsymbol{\mu}_0 \in \mathbb{R}^{D+K}$ and diagonal $\boldsymbol{\Sigma_0} = \boldsymbol{I}\sigma_0^2$.

The Bernoulli process is then used by each local model to select a subset of global neurons:$\mathcal{T}_s = \sum_i a_{si}\delta_{\boldsymbol{\theta}_i}|M \sim \text{BeP}(M)$ for $s = 1, \ldots, S$, where $a_{si}|m_i \sim \text{Bern}(m_i)$ is a random measure representing a subset of global neurons contained in local model s. Considering the inherent permutation invariance of local model, it denotes $\boldsymbol{A} = \{A_{ij}^s\}_{s,i,j}$ as the assignment variables, $\sum_j A_{ij}^s = a_{si} \in \{0, 1\}$, $\sum_i A_{ij}^s = 1$. Finally, observed local neurons in model s are treated as noisy measurements of global neurons under permutation invariance:

$$\boldsymbol{w}_{sj} \mid \boldsymbol{A}^s, \boldsymbol{\theta} \sim \mathcal{N}(\sum_i A_{ij}^s\boldsymbol{\theta}_i, \boldsymbol{\Sigma_s}) \text{ for } j = 1, \ldots, J_s, \quad (1)$$

where $J_s = card(\mathcal{T}_s)$, $\boldsymbol{\Sigma_s} = \boldsymbol{I}\sigma_s^2$ is also diagonal and represents the noise. The noise is usually caused by estimation error due to finite sample sizes or variations in the distribution of each local dataset. $A_{ij}^s = 1$ indicates that $\boldsymbol{w}_{sj}$ is matched to $\boldsymbol{\theta}_i$, i.e. $\boldsymbol{w}_{sj}$ is the local neuron realization of the global neuron $\boldsymbol{\theta}_i$; $A_{ij}^s = 0$ indicates the inverse.

After the modeling, it is natural to infer the global neurons by maximizing the posterior probability of the global neurons given local neurons (likelihoods) under the permutation invariance:

$$\begin{aligned}\max_{\{\boldsymbol{\theta}_i\},\{\boldsymbol{A}^s\}} & P(\{\boldsymbol{\theta}_i\}, \{\boldsymbol{A}^s\}|\{\boldsymbol{w}_{sj}\}) \propto \\ & P(\{\boldsymbol{w}_{sj}\}|\{\boldsymbol{\theta}_i\}, \{\boldsymbol{A}^s\})P(\{\boldsymbol{A}^s\})P(\{\boldsymbol{\theta}_i\}),\end{aligned} \quad (2)$$

define $\boldsymbol{Z}_i = \{(s, j)|A_{ij}^s = 1\}$ be the index set of local neurons assigned to i-th global neuron, and taking negative logarithm it can obtain:

$$\begin{aligned}\min_{\{\boldsymbol{\theta}_i\},\{\boldsymbol{A}^s\}} & -\log(P(\{\boldsymbol{A}^s\})) \\ & -\sum_i\Big(\sum_{z\in\boldsymbol{Z}_i}\log(p(\boldsymbol{w}_z|\boldsymbol{\theta}_i)) + \log(p(\boldsymbol{\theta}_i))\Big)\end{aligned} \quad (3)$$

where $P(\{\boldsymbol{A}^s\})$ is interpreted by Indian Buffet Process (IBP) and demonstrated in Appendix B. Given $\{\boldsymbol{A}^s\}_{s=1}^S$, the closed form of $\{\boldsymbol{\theta}_i\}$ can be estimated through the Gaussian-Gaussian conjugacy:

$$\boldsymbol{\theta}_i = \frac{\boldsymbol{\mu}_0/\sigma_0^2 + \sum_{s,j}\boldsymbol{A}_{i,j}^s\boldsymbol{w}_{sj}/\sigma_s^2}{1/\sigma_0^2 + \sum_{s,j}\boldsymbol{A}_{i,j}^s/\sigma_s^2} \text{ for } i = 1, \cdots, J. \quad (4)$$

Taking equation (4) into objective (3), it can cast optimization only with respect to $\{\boldsymbol{A}^s\}_{s=1}^S$. Unfortunately, solving all local assignments together leading an NP-hard combinatorial optimization problem. Thus, PFNM applies an alternative optimization process to solve the problem.

Alternative Optimization PFNM iteratively optimizes one local assignment variable $\boldsymbol{A}^{s'}$ at a time by fixing all other assignment variables $\{A_{i,j}^s\}_{i,j,s\in -s'}$, where $-s'$ denotes "all but s'". It aims to formulate a linear sum assignment problem with current local assignment $\sum_{i,j} C_{i,j}^{s'}A_{i,j}^{s'}$ in each iteration, where $\{C_{i,j}^{s'}\}_{i,j}$ denotes the cost specification. The Hungarian algorithm is then used to solve the

linear sum assignment problem. In each iteration, it divides terms in objective (3) into two parts: one is $i = 1, \cdots, J_{-s'}$, where $J_{-s'} = \max\{i : \boldsymbol{A}^s_{ij} = 1, \text{for } s \in -s', j = 1, \ldots, J_s\}$, this part denotes active global neurons estimated from $-s'$; another is $i = J_{-s'} + 1, \cdots, J_{-s'} + J_{s'}$, it denotes new global neuron from current local model. Then it has the following proposition describes the assignment cost $\{C^{s'}_{i,j}\}_{i,j}$:

Proposition 1. *The assignment cost specification $C^{s'}_{i,j}$ for finding $\{\boldsymbol{A}^{s'}\}$ is $C^{s'}_{i,j} =$*

$$\begin{cases} 2\log\frac{S-n_i^{-s'}}{n_i^{-s'}} - \frac{\left\|\frac{\boldsymbol{\mu}_0}{\sigma_0^2} + \frac{\boldsymbol{w}_{s'j}}{\sigma^2_{s'}} + \sum_{s\in-s',j}\boldsymbol{A}^s_{i,j}\frac{\boldsymbol{w}_{sj}}{\sigma_s^2}\right\|^2}{\frac{1}{\sigma_0^2}+\frac{1}{\sigma^2_{s'}}+\sum_{s\in-s',j}\boldsymbol{A}^s_{i,j}\frac{1}{\sigma_s^2}} \\ \qquad + \frac{\left\|\frac{\boldsymbol{\mu}_0}{\sigma_0^2} + \sum_{s\in-s',j}\boldsymbol{A}^s_{i,j}\frac{\boldsymbol{w}_{sj}}{\sigma_s^2}\right\|^2}{\frac{1}{\sigma_0^2}+\sum_{s\in-s',j}\boldsymbol{A}^s_{i,j}\frac{1}{\sigma_s^2}}, & i \le J_{-s'} \\ 2\log\frac{i-J_{-s'}}{\gamma_0/S} - \frac{\left\|\frac{\boldsymbol{\mu}_0}{\sigma_0^2} + \frac{\boldsymbol{w}_{s'j}}{\sigma^2_{s'}}\right\|^2}{\frac{1}{\sigma_0^2}+\frac{1}{\sigma^2_{s'}}} \\ \qquad + \frac{\left\|\frac{\boldsymbol{\mu}_0}{\sigma_0^2}\right\|^2}{\frac{1}{\sigma_0^2}}, & J_{-s'} < i \le J_{-s'} + J_{s'}, \end{cases} \tag{5}$$

where $n_i^{-s'} = \sum_{s\in-s',j}\boldsymbol{A}^s_{i,j}$ denotes the number of local neurons were assigned to global neuron i outside of s'.

The proof can be found in Appendix B and (Yurochkin et al. 2019b).

Analysis of PFNM

Before we discuss, we have the following proposition:

Proposition 2. *For any prior of a global neuron $\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{I}\sigma_0^2)$, when it is assigned with a likelihood observed local neuron $\boldsymbol{w}_{s'j}|\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\theta}_i, \boldsymbol{I}\sigma^2_{s'})$, the posterior distribution of this global neruon is $\boldsymbol{\theta}_i|\boldsymbol{w}_{s'j} \sim \mathcal{N}\Big(\frac{\boldsymbol{\mu}_0/\sigma_0^2+\boldsymbol{w}_{s'j}/\sigma^2_{s'}}{1/\sigma_0^2+1/\sigma^2_{s'}}, \boldsymbol{I}\big(\frac{1}{1/\sigma_0^2+1/\sigma^2_{s'}}\big)\Big)$; when it has been assigned local neurons from other assignments $\boldsymbol{Z}_i^{-s'} = \{(s,j)|A^s_{ij} = 1, s \in -s'\}$, the posterior distribution is $\boldsymbol{\theta}_i|\boldsymbol{Z}_i^{-s'} \sim \mathcal{N}(\frac{\boldsymbol{\mu}_0/\sigma_0^2+\sum_{s\in-s',j}A^s_{i,j}\boldsymbol{w}_{sj}/\sigma_s^2}{1/\sigma_0^2+\sum_{s\in-s',j}A^s_{i,j}/\sigma_s^2}, \boldsymbol{I}(\frac{1}{1/\sigma_0^2+\sum_{s\in-s',j}A^s_{i,j}/\sigma_s^2}))$.*

The proof can be found in Appendix C.

Terms of equation (5) on the left are due to $P(\boldsymbol{A}^{s'}|\{\boldsymbol{A}^{-s'}\})$. At initial stage of iteration all $n_i^{-s'}$ are identical. So the main differences compared by cost $C^{s'}_{i,j}$ induced from right term.

Definition 3. *(Standardized mean square) For any multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}\sigma^2)$, we call $\|\boldsymbol{\mu}\|^2/\sigma^2$ as standardized mean square of this distribution, and denote it as $\mathbb{S}(\mathcal{N}) = \|\boldsymbol{\mu}\|^2/\sigma^2$.*

The proposition 2 shows that when $i \le J_{-s'}$, the right term is the difference in standardized mean square between distribution $p(\boldsymbol{\theta}_i|\boldsymbol{Z}_i^{-s'})$ and this distribution assigned with local likelihood $p(\boldsymbol{\theta}_i|\boldsymbol{Z}_i^{-s'}, \boldsymbol{w}_{s'j})$; when $J_{-s'} < i \le J_{-s'} + J_{s'}$, the right term is the difference of standardized mean square between prior distribution $p(\boldsymbol{\theta}_i)$ and this distribution assigned with local likelihood $p(\boldsymbol{\theta}_i|\boldsymbol{w}_{s'j})$. So, by comparing each cost term $C^{s'}_{i,j}$, it essentially compares the standardized mean square difference between the i-th global neuron distribution (either estimated from $-s'$ or a new one from the current local model) and this distribution assigned with local neuron $w_{s'j}$. However, we have the following proposition shows the issue caused by the cost term:

Proposition 4. *For a global neuron distribution $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{I}\sigma_0^2)$, after it is assigned with a likelihood $p(\boldsymbol{w}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{I}\sigma^2)$, the standardized mean square difference between $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{w})$ is proportion to $\|\boldsymbol{w}\|^2/\sigma^2$.*

Proof. Denote the mean and covariance matrix of $p(\boldsymbol{\theta}|\boldsymbol{w})$ as $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{I}\tilde{\sigma}^2$. From definition 3, we can write the the standardized mean square difference between $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{w})$ as

$$\begin{aligned} &\mathbb{S}(p(\boldsymbol{\theta})) - \mathbb{S}(p(\boldsymbol{\theta}|\boldsymbol{w})) \\ =&\frac{\|\boldsymbol{\mu}_0\|^2}{\sigma_0^2} - \frac{\|\tilde{\boldsymbol{\mu}}\|^2}{\tilde{\sigma}^2} \\ =&\log\frac{\exp\big(\frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\boldsymbol{\mu}_0}{2\sigma_0^2}\big)}{(2\pi\sigma_0^2)^{\frac{D+K}{2}}p(\boldsymbol{\theta})} - \log\frac{\exp\big(\frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\tilde{\boldsymbol{\mu}}}{2\tilde{\sigma}^2}\big)}{(2\pi\tilde{\sigma}^2)^{\frac{D+K}{2}}p(\boldsymbol{\theta}|\boldsymbol{w})} \\ =&\log\frac{\exp\big(\frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\boldsymbol{\mu}_0}{2\sigma_0^2}\big)}{(2\pi\sigma_0^2)^{\frac{D+K}{2}}p(\boldsymbol{\theta})} \cdot \frac{(2\pi\tilde{\sigma}^2)^{\frac{D+K}{2}}p(\boldsymbol{\theta}|\boldsymbol{w})}{\exp\big(\frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\tilde{\boldsymbol{\mu}}}{2\tilde{\sigma}^2}\big)}. \end{aligned} \tag{6}$$

Because

$$\begin{aligned} &\Big(\frac{\tilde{\sigma}^2}{\sigma_0^2}\Big)^{\frac{D+K}{2}} \cdot \exp\Big(\frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\boldsymbol{\mu}_0}{2\sigma_0^2} - \frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\tilde{\boldsymbol{\mu}}}{2\tilde{\sigma}^2}\Big) \\ =&\Big(\frac{\tilde{\sigma}^2}{\sigma_0^2}\Big)^{\frac{D+K}{2}} \cdot \exp\Big(\boldsymbol{\theta}^T\boldsymbol{\theta}\Big(-\frac{1}{2\sigma_0^2}+\frac{1}{2\tilde{\sigma}^2}\Big) + \boldsymbol{\theta}^T\Big(\frac{\boldsymbol{\mu}_0}{\sigma_0^2} - \frac{\tilde{\boldsymbol{\mu}}}{\tilde{\sigma}^2}\Big)\Big), \end{aligned} \tag{7}$$

from proposition 2, we can obtain

$$\frac{1}{2}\Big(-\frac{1}{\sigma_0^2}+\frac{1}{\tilde{\sigma}^2}\Big) = \frac{1}{2\sigma^2}, \qquad \frac{\boldsymbol{\mu}_0}{\sigma_0^2} - \frac{\tilde{\boldsymbol{\mu}}}{\tilde{\sigma}^2} = -\frac{\boldsymbol{w}}{\sigma^2}, \tag{8}$$

$\Big(\frac{\tilde{\sigma}^2}{\sigma_0^2}\Big)^{\frac{D+K}{2}} = \big(1+\frac{\sigma_0^2}{\sigma^2}\big)^{-\frac{D+K}{2}}$ is a constant in PFNM setting, and from Bayesian theory

$$p(\boldsymbol{\theta}|\boldsymbol{w}) \propto p(\boldsymbol{\theta})p(\boldsymbol{w}|\boldsymbol{\theta}), \tag{9}$$

thus we have

$$\begin{aligned} \mathbb{S}(p(\boldsymbol{\theta})) - \mathbb{S}(p(\boldsymbol{\theta}|\boldsymbol{w})) &\propto \log\frac{p(\boldsymbol{w})}{\exp\big(\frac{-\boldsymbol{\theta}^T\boldsymbol{\theta}+2\boldsymbol{\theta}^T\boldsymbol{w}}{2\sigma^2}\big)} \\ &\propto -\frac{\boldsymbol{w}^T\boldsymbol{w}}{2\sigma^2}. \end{aligned} \tag{10}$$

□

From proposition 4, We can know that the differences compared by each $C_{ij}^{s'}$ are only related to local neuron. As a consequence, for one fixed local neuron $\boldsymbol{w}_{s'j}$ and two different global neurons $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_{i^*}$, the cost does not discriminate global neurons $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_{i^*}$. Hence, the cost specifications induced from original PFNM can't find optimal solutions for $\{\boldsymbol{A}^{s'}\}$.

Kullback-Leibler Divergence

From the above analysis, we know that the cost specification $C_{i,j}^{s'}$ induced from PFNM omits information of global neuron, i.e. the i index in the cost. So, it is natural to fix the issue by adding a regularized term that contains both information of i and j. As we point out before, the cost term $C_{i,j}^{s'}$ essentially is the standardized mean square difference between i-th global distribution and this distribution assigned with local neuron $w_{s'j}$. From the probabilistic perspective, minimizing the cost is meant to hope these two distributions are as close as possible. So how to measure the distance between two distributions? Kullback-Leibler becomes the first answer that gets into mind. Can KL-divergence fix the issues caused by PFNM? To answer this question, we have the following proposition:

Proposition 5. *For a global neuron distribution $p(\boldsymbol{\theta})$, after it is assigned with a likelihood $p(\boldsymbol{w}|\boldsymbol{\theta})$, the Kullback–Leibler Divergence Penalty between $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{w})$ is equal up to $\mathbb{E}_{p(\boldsymbol{\theta})}\big[\log p(\boldsymbol{w} \mid \boldsymbol{\theta})\big]$.*

Proof.

$$\begin{aligned}&\mathrm{KL}\big(p(\boldsymbol{\theta})\|p(\boldsymbol{\theta}|\boldsymbol{w})\big)\\ &=\int p(\boldsymbol{\theta})\log\frac{p(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{w})}\,\mathrm{d}\boldsymbol{\theta}\end{aligned} \tag{11}$$

and from Bayesian theory

$$p(\boldsymbol{\theta}|\boldsymbol{w}) \propto p(\boldsymbol{\theta})p(\boldsymbol{w}|\boldsymbol{\theta}), \tag{12}$$

so we have

$$\mathrm{KL}\big(p(\boldsymbol{\theta})\|p(\boldsymbol{\theta}|\boldsymbol{w})\big) \cong -\mathbb{E}_{p(\boldsymbol{\theta})}\big[\log p(\boldsymbol{w} \mid \boldsymbol{\theta})\big] \tag{13}$$

□

From above proposition, KL divergence between two distribution is essentially equal up to the expectation of the assigned local likelihood distribution, and the expectation is taken over the global distribution $p(\boldsymbol{\theta})$. That means the KL divergence contains both information of local and global neuron.

Thus, to fix the drawback induced from PFNM, we can formulate a new cost specifications that regularize original cost specification with the KL penalty, i.e., $\widetilde{\boldsymbol{C}}_{i,j}^{s'} =$

$$\begin{cases}\boldsymbol{C}_{i,j}^{s'}+\lambda\,\mathrm{KL}\Big(p(\boldsymbol{\theta}_i|\boldsymbol{Z}_i^{-s'})\big\|p(\boldsymbol{\theta}_i|\boldsymbol{Z}_i^{-s'},\boldsymbol{w}_{s'j})\Big), i\le J_{-s'},\\ \boldsymbol{C}_{i,j}^{s'}+\lambda\,\mathrm{KL}\Big(p(\boldsymbol{\theta}_i)\big\|p(\boldsymbol{\theta}_i|\boldsymbol{w}_{s'j})\Big), J_{-s'}<i\le J_{-s'}+J_{s'}.\end{cases} \tag{14}$$

where the coefficient λ is the adjusting ratio. And the KL divergence between two multivariate normal distributions can be calculated by the following Lemma:

Lemma 5.1. *The Kullback-Leibler divergence between $\mathcal{N}_x(\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)$ and $\mathcal{N}_y(\boldsymbol{\mu}_y,\boldsymbol{\Sigma}_y)$, is:*

$$\frac{1}{2}[\mathrm{tr}(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_x)+(\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)+dim+\ln\frac{|\boldsymbol{\Sigma}_y|}{|\boldsymbol{\Sigma}_x|}], \tag{15}$$

where $dim = D + K$ is the dimension of $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$.

The proof can be found in (Duchi 2007).

We call our new method as neural aggregation with full information (NAFI). And the process of NAFI can be summarized in algorithm 1.

As demonstrated by the study in (Wang et al. 2020), directly applying the matching algorithms fails on deep architectures designed for more complex tasks. To address this issue, we extend NAFI to a layer-wise matching scheme, which can be found in Appendix D.

Algorithm 1: The process of NAFI

Require:
Local weights $\boldsymbol{w}_{sj}$ from S local models;
Ensure:
Matching assignments $\{\boldsymbol{A}^s\}_{s=1}^S$, global weights $\{\boldsymbol{\theta}_i\}$;
for t = 1, 2, 3, $\cdots$ **do**
  for s' = 1, 2, $\cdots$, S **do**
    Giving assignments $\{\boldsymbol{A}_{i,j}^s\}_{i,j,s\in -s'}$, acquiring corresponding global model via equations (4): $\tilde{\boldsymbol{\theta}}_i = \frac{\boldsymbol{\mu}_0/\sigma_0^2+\sum_{s\in-s',j}\boldsymbol{A}_{i,j}^s\boldsymbol{w}_{sj}/\sigma_s^2}{1/\sigma_0^2+\sum_{s\in-s',j}\boldsymbol{A}_{i,j}^s/\sigma_s^2}$;
    Acquire the assignment cost $\{\widetilde{\boldsymbol{C}}_{i,j}^{s'}\}_{i,j}$ via equation (14);
    Obtain $\{\boldsymbol{A}_{i,j}^{s'}\}_{i,j}$ by solving the linear assignment problem via Hungarian algorithm.
  end for
  Apply $\{\boldsymbol{A}^s\}_{s=1}^S$ to update the global model.
end for

Experiments

To evaluate the efficiency of the proposed NAFI, we presents an empirical study of NAFI and compares it with PFNM, FedAvg (McMahan et al. 2017), FedProx (Li et al. 2018) and optimal transport fusion (Singh and Jaggi 2020) (which we refer as OT fusion in remaining part). The experiments are carried out over three datasets with three types of neural networks: FCNN, shallow CNN and U-net. The experiments below indicate that our algorithm can aggregate multiple neural networks into a more efficient global neural network[1].

Datasets, models and metrics Our algorithm is evaluated on three datasets: MINST, CIFAR 10 and Carvana Image Masking Challenge (CIMC). MINST and CIFAR-10 are image classification datasets and each contains ten classes on handwriting digits and objects in real life respectively. CIMC is a binary semantic segmentation dataset consisting of photos of cars, and the task is to split out the car and

[1]https://github.com/XiaoPeng24/NAFI

the background. For MNIST, we use an FCNN model and evaluate it with accuracy; for CIFAR 10, we use a ConvNet with 3 convolutional and 2 fully-connected layers and evaluate it with accuracy; and for CIMC, we use the U-net (Ronneberger, Fischer, and Brox 2015) and evaluate it with dice coefficient.

Partition strategies of local data To simulate a federated learning scenario, we partition each dataset in a heterogeneous strategy in which the number of data points and class proportions in each local model is unbalanced. We follow prior works (Yurochkin et al. 2018) in the heterogeneous partition of local data for three datasets, which apply K-dimensional Dirichlet distribution $Dir(\alpha)$ to create non-identical independent distribution (iid) data, with a smaller α indicating higher data heterogeneity. Specifically, for dataset with class number K, we sample the proportion of the instances of class k to local model s, $p_{k,s}$, via $p_{k,s} \sim Dir_k(\alpha)$. For MNIST and CIFAR10, $K = 10$. For CIMC, $K = 1$ as it is a binary semantic segmentation dataset. In each dataset, we execute 5 trials to obtain the mean and standard deviation of the performances.

Baselines Our method is compared to the original PFNM, FedAvg, OT fusion and FedProx. FedAvg and FedProx are executed in local neural networks that have been trained using the same random initialization as proposed by (McMahan et al. 2017). We note that while a federated averaging variant without shared initialization is likely to be more realistic when attempting to aggregate pre-trained models, it performs significantly worse than all other baselines. In CNN architectures, the original PFNM we compared is actually its deep model extensive version—FedMA (Wang et al. 2020). Considering the high heterogeneity in each trial, we set λ in a wide grid (from 10^{-8} to 1) and choose the best result for NAFI. We subsequently check the sensitivity of λ and find it only have tiny fluctuation.

	MNIST	CIFAR-10	CIMC
Model	FCNN	ConvNet	U-net
Optimizer	Adam	SGD	RMSprop
Learning rate	0.01	0.01	0.01
Size of minibatch	32	32	3
Epochs	10	10	3

Table 1: Hyperparameter settings for training neural networks.

Training setup We use PyTorch (Paszke et al. 2019) to implement these networks and train them by the Adam (Kingma and Ba 2014), SGD (Bottou 2010) and RMSprop (Hinton, Srivastava, and Swersky 2012) with hyperparameter settings which are summarized in Table 1.

Performance Overview In real world applications of Federated Learning, the discrepancy of data distribution among local models and communication cost will inevitably increase as the number of local models increases. This is mainly due to the variability of data generation paradigms in the system (Li et al. 2020). This suggests that testing fusing algorithms for various numbers of local models S is important. For MNIST, we firstly apply various baselines with 15, 20, 25 and 30 local models with one hidden layer FCNN. Local NN in Table 2 reports the average of separately tested network accuracies. The lower extremes of aggregating are determined by the performance of local NNs. As shown in Table 2, accuracy of NAFI on MNIST is 4% higher than PFNM on average. We also examine how the performance of all methods is impacted by the number of hidden layers N. We train 10 neural networks with 2 and 3 hidden layers respectively and then fuse them using various baselines. The performance of PFNM decreases with the increasing number of hidden layers—even worse than OT fusion. This phenomenon has already been proposed in (Wang et al. 2020), and it can be fixed by a layer-wise training manner which we apply in CNN architectures. It is worth mentioning that NAFI seems have the potential to address the drawback of PFNM in deep architecture, as NAFI still maintain at a moderate level in deep architecture. This may be explained by an accumulating error effect. From above analysis, we discovered that PFNM is unable to discriminate those global neurons because the cost specification does not include global neuron information, while NAFI equipped with KL divergence can correct this. Thus, with the fusing process of neural networks is going layer-by-layer, the drawback of PFNM described above takes effect, and global neurons are not generated correctly. As a result, the incorrectness of PFNM in picking global neurons is superimposed, and the performance disparity between PFNM and NAFI is amplified as presented in Table 2.

For CNN application, we apply ConvNet (2 convolutional layers and 3 fully connected layers) on 5, 10, 15, and 20 local models and use various methods to fuse the models trained in CIFAR 10's heterogeneous dataset partition. We apply PFNM and NAFI in an iteratively layer-wise way mentioned above, so we also set the communications rounds of FedAvg, FedProx and OT fusion to 5 equal to the number of layers in ConvNet for equality. As shown in Table 3, NAFI outperforms the other methods on fusing convolutional neural networks. Concretely, the accuracy of NAFI on CIFAR 10 is 3% higher than PFNM on average for a various number of local models.

U-Net (Ronneberger, Fischer, and Brox 2015) is widely used on medical image segmentation (Du et al. 2020), and medical images such as tumor scan images generated by multiple institutions are generally forbidden to be exchanged for privacy issues. Therefore, it is critical to federate well-known image segmentation algorithms. To this end, we compare our method and baselines on U-Net. The U-Net architecture is described in (Ronneberger, Fischer, and Brox 2015), and it employs the batch normalization layer, which is not extended by the matching type federated learning algorithm yet. Here we apply a technique which fuses batch normalization layer into convolutional layer, and extend the matching type algorithm to batch normalization layer. The details can be referred in Appendix E. As shown in Table 4, FedAvg, FedProx and OT fusion have very subpar perfor-

	S	N	Local NN	FedAvg	FedProx	OT fusion	PFNM	NAFI
nets	15	1	71.9 ± 2.57	75.47 ± 5.90	75.65 ± 5.93	81.44 ± 3.21	83.93 ± 0.14	$\mathbf{87.34 \pm 0.57}$
	20	1	69.44 ± 2.50	75.02 ± 4.03	75.17 ± 3.99	82.71 ± 3.57	83.23 ± 3.31	$\mathbf{86.73 \pm 2.15}$
	25	1	67.99 ± 2.00	75.46 ± 3.33	75.24 ± 3.30	82.53 ± 3.05	84.87 ± 1.66	$\mathbf{86.46 \pm 2.56}$
	30	1	65.90 ± 2.66	73.43 ± 4.49	73.11 ± 4.44	82.74 ± 2.41	83.39 ± 2.23	$\mathbf{86.23 \pm 2.57}$
layers	10	2	73.21 ± 3.05	66.56 ± 6.82	66.34 ± 6.75	83.77 ± 4.23	79.09 ± 4.73	$\mathbf{84.58 \pm 4.54}$
	10	3	70.28 ± 2.97	52.01 ± 6.52	51.79 ± 6.49	67.56 ± 5.64	60.71 ± 4.59	$\mathbf{72.04 \pm 4.75}$

Table 2: Test accuracy of FCNN in MNIST.

	S	Local NN	FedAvg	FedProx	OT fusion	PFNM (FedMA)	NAFI
nets	5	25.21 ± 2.30	51.26 ± 2.97	51.43 ± 3.01	52.12 ± 2.78	50.39 ± 0.94	$\mathbf{52.56 \pm 0.81}$
	10	18.92 ± 1.41	46.78 ± 2.36	46.94 ± 2.32	45.43 ± 3.45	46.27 ± 1.73	$\mathbf{47.46 \pm 1.42}$
	15	15.85 ± 0.51	42.40 ± 2.25	42.06 ± 2.20	40.79 ± 4.29	42.82 ± 2.46	$\mathbf{45.72 \pm 1.35}$
	20	14.11 ± 0.53	34.20 ± 2.54	34.02 ± 2.52	37.56 ± 4.13	42.61 ± 2.07	$\mathbf{44.96 \pm 1.93}$

Table 3: Test accuracy of Convnet in CIFAR10.

mances. And FedAvg and FedProx even cannot converge to a stationary result in such complex network architecture (the results go up and down during communication rounds). The result shown in the table is taken from the best result during all 19 communication rounds which are equal to the number of layers in U-net. As shown in Table 4, the priority of NAFI is significantly more important among popular Federated Learning methods than it is in the case of 2- or 3-layer neural networks. With 8 local models, NAFI on CIMC has accuracy that is at least 7% higher than PFNM, and accuracy that is almost 8% higher with 16 local models.

Amount of fused global model In real federated application, the amount of fused global model determines the burden of communication. Hence we also measure the size of global model fused by PFNM and NAFI respectively. In figure 1, we plot the logarithm ratio between the amount of fused global and the total amount of all local models $\log \frac{J}{\sum_s J_s}$ in FCNN multiple nets scenario. As shown in figure 1, the logarithm ratio decrease with the increase in number of local models and is kept lower than -0.5, which means the matching type method produces a relative small model while can capture the most advantages of local models. It is worth noting that the ratio of NAFI is always small than PFNM, while the performance of fused global model in NAFI outperforms than PFNM accordint to above experiment results. This indicates the availability of NAFI in some model compression tasks.

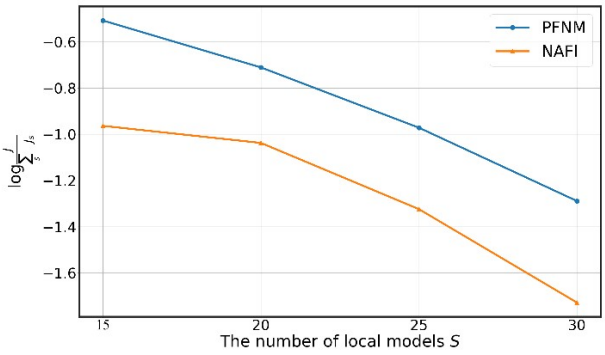


Figure 1: The logarithm ratio between the amount of fused global and the total amount of all local models in FCNN multiple nets scenario.

Sensitivity Analysis

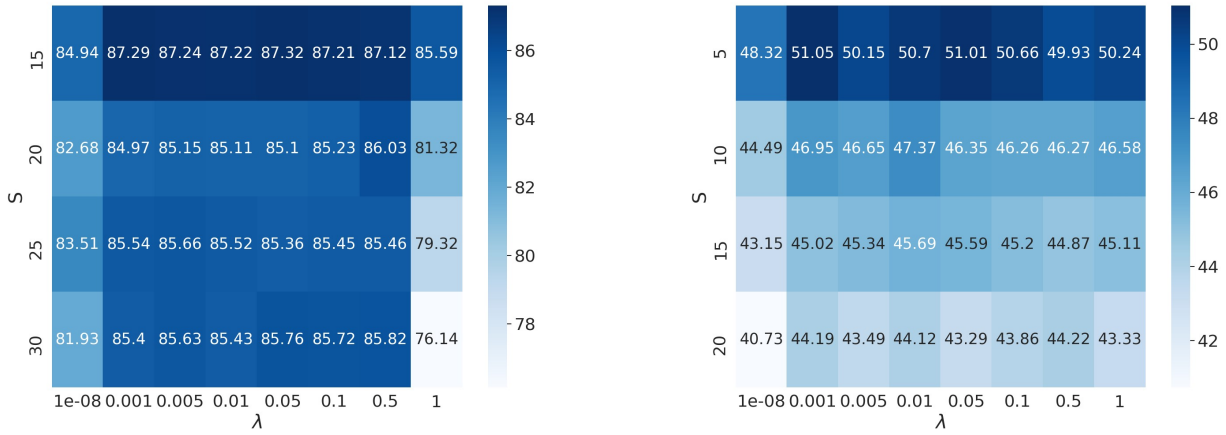
The weight of the KL-divergence term, λ, is a hyperparameter that we introduce into NAFI. Therefore, it is essential to demonstrate the sensitivity of λ. Here, we set λ to various positive values for NAFI applied on fusing FCNNs and CNNs trained in MNIST and CIFAR10 respectively. We also take the impact of the number of local models into account. As shown in figure 2, the heat map indicates the accuracy on the test data, and for various number of clients, there is only a tiny fluctuation of prediction accuracy for a fused global model when $10^{-8} \leq \lambda \leq 1$. Although NAFI with $\lambda = 10^{-8}$ performs significantly worse than NAFI with $10^{-3} \leq \lambda \leq 0.5$, the fused global model maintains a high level of performance. When $\lambda = 1$, the performance of the fused global model drops sharply, this indicates that λ should not set too high. To summarize, NAFI is robust on the hyperparameter λ under a variety of conditions.

Conclusion

In this study, we theoretically analyze the flaw of PFNM and propose a new federated neural matching method to fix this flaw. It is empirically shown that the new method outperforms other state-of-the-art algorithms for federated learning of neural networks, and the comparison of fused global model size also indicates the potential availability of our

	S	Local NN	FedAvg	FedProx	OT fusion	PFNM (FedMA)	NAFI
nets	8	67.60 ± 9.38	53.28 ± 10.63	41.53 ± 0.82	63.57 ± 5.48	90.31 ± 2.90	**96.47 ± 0.76**
	16	27.84 ± 12.16	42.62 ± 0.87	48.50 ± 9.52	54.49 ± 5.62	75.47 ± 3.54	**83.46 ± 3.41**

Table 4: Test dice coefficient of U-net in CIMC.



(a) MNIST multiple nets

(b) CIFAR10 multiple nets

Figure 2: KL regularization coefficients sensitivity analysis

method in model compression and pruning tasks. In future work, we will extend our algorithm into more advanced architectures, like transformers etc. Additionally, the success of KL-divergence suggests it is also likely to try other metrics which can measure the similarity between two probability distributions.

Acknowledgements

During the process of preparing this paper, the FIFA World Cup Qatar 2022 is underwaying. The Argentina football player Lionel Messi and his Argentina men's football team have also successfully qualified for the group stage. For a long time, Messi has maintained a brilliant and lasting career through his own efforts and self-discipline. This kind of hard work and self-discipline also deeply moved the author, and Messi also become the spiritual mentor of the author during his career path. Lack of a World Cup trophy is the only barrier for Messi being the world's number one football player. In 2014, Messi regretted passing it by. Here, the author wishes Messi to win the World Cup trophy in this year through his effort. And all of us should believe that God rewards those who work hard. Let's work hard for our goals.

References

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.

Buciluǎ, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 535–541.

Chen, H.-Y.; and Chao, W.-L. 2020. Fedbe: Making Bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*.

Claici, S.; Yurochkin, M.; Ghosh, S.; and Solomon, J. 2020. Model Fusion with Kullback–Leibler Divergence. *arXiv preprint arXiv:2007.06168*.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2021. Distributionally robust federated averaging. *arXiv preprint arXiv:2102.12660*.

Du, G.; Cao, X.; Liang, J.; Chen, X.; and Zhan, Y. 2020. Medical image segmentation based on U-net: A review. *Journal of Imaging Science and Technology*, 64(2): 20508–1.

Duchi, J. 2007. Derivations for linear algebra and optimization. *Berkeley, California*, 3(1): 2325–5870.

Hinton, G.; Srivastava, N.; and Swersky, K. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8): 2.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*.

Lin, T.; Kong, L.; Stich, S. U.; and Jaggi, M. 2020. Ensemble distillation for robust model fusion in federated learning. *arXiv preprint arXiv:2006.07242*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic Federated Learning. In *International Conference on Machine Learning*, 4615–4625.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Schmidhuber, J. 1992. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2): 234–242.

Singh, S. P.; and Jaggi, M. 2020. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33.

Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 4424–4434.

Thibaux, R.; and Jordan, M. I. 2007. Hierarchical Beta Processes and the Indian Buffet Process. *Journal of Machine Learning Research*, 2(3): 564–571.

Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*.

Xiao, P.; Cheng, S.; Stankovic, V.; and Vukobratovic, D. 2020. Averaging is probably not the optimum way of aggregating parameters in federated learning. *Entropy*, 22(3): 314.

Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; and Hoang, N. 2019a. Statistical model aggregation via parameter matching. *Advances in Neural Information Processing Systems*, 32: 10956–10966.

Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; Hoang, N.; and Khazaeni, Y. 2019b. Bayesian Nonparametric Federated Learning of Neural Networks. In *International Conference on Machine Learning*, 7252–7261.

Yurochkin, M.; Fan, Z.; Guha, A.; Koutris, P.; and Nguyen, X. 2018. Scalable inference of topic evolution via models for latent geometric structures. *arXiv preprint arXiv:1809.08738*.

Zhang, X.; Hong, M.; Dhople, S.; Yin, W.; and Liu, Y. 2020. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418*.

Zhu, Z.; Hong, J.; and Zhou, J. 2021. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. *arXiv preprint arXiv:2105.10056*.