Approaches to Uncertainty Quantification in Federated Deep Learning

Florian Linsner[1], Linara Adilova[1], Sina Däubener[1], Michael Kamp[2], and Asja Fischer[1]

[1] Faculty of Mathematics, University of Bochum <name>.<surname>@rub.de
[2] Dept of Data Science and AI, Faculty of IT, Monash University
michael.kamp@monash.edu

Abstract. Trustworthy machine learning allows data privacy and a robust assessment of the uncertainty of predictions. Methods for quantifying uncertainty in deep learning have recently gained attention, while federated deep learning allows to utilize distributed data sources in a privacy-preserving manner. In this paper, we integrate several approaches for uncertainty quantification in federated deep learning. In particular, we show that prominent approaches such as MC-dropout and stochastic weight averaging Gaussian (SWAG) can be extended efficiently to federated setup. Moreover, we demonstrate that deep ensembles allow for natural integration in the federated learning framework. Our empirical evaluation confirms that a trustworthy uncertainty quantification on out-of-distribution data is possible in federated learning with little (SWAG) to no (MC-dropout, ensembles) additional communication. While all methods perform well in our empirical analysis and should serve as baselines in future developments in this field, deep ensembles and MC-dropout allow for better uncertainty based identification of out-of-distribution data and wrong classified data.

Keywords: Federated deep learning · Uncertainty · OOD detection.

1 Introduction

Deep learning is used in many critical application areas, such as healthcare [47, 44] or autonomous driving [29, 19].

However, predictions of deep learning models can be mistaken, especially on unseen data. When machine learning is applied as a tool, not minding the possible flaws of the models can be very costly, e.g., for medical diagnosis or nuclear power plants control decisions. In such critical environments it is important that a model can quantify the certainty of its predictions. While for classification an uncertainty score can be derived from the widely used softmax output, it is often uncalibrated and over-confident or misleading [13].

To obtain reliable uncertainty estimates, multiple methods for uncertainty quantification for deep learning models have been proposed [34, 8, 26, 23, 32, 28, 10]. Bayesian neural networks dating back to Neal [36] offer a natural way

of quantifying uncertainty by marginalizing over the parameter posterior distribution. Moreover, the induced stochasticity can be used for evaluating the uncertainty of the prediction in terms of prediction variance [9]. While Neal [36] introduced the Hamiltonian Monte Carlo method for deriving the posterior, simultaneously MacKay [31] analyzed methods based on the Laplace approximation. Both approaches have been further extended for scalability [41, 46], however challenges like the convergence of the Markov chain in the first, and calculating the Hessian in the second remain. Another line of research uses approximate variational inference [15, 12], where a simpler parametrized distribution is fit to approximate the true posterior by maximizing the variational lower bound of the log likelihood, also referred to as evidence lower bound (ELBO). For better scalability practical approximate Bayesian implementations like MC-dropout [10] or SWAG [32] have been developed, as well as more empirical approaches such as deep ensembles [28].

With increasing amounts of data, the overhead of quantifying uncertainty can be challenging when training a neural network. For the efficient training of neural networks on large-scale distributed datasets, various parallelization methods have been proposed. On-device, edge, or in-situ processing, has been well studied in the context of stream processing [42, 11] and monitoring functions over sensor networks [5, 24, 7]; averaging models have been used in online learning from distributed data streams [20, 22]. For deep learning, training models in-situ and averaging their parameters on a coordinator node was termed *federated learning* [25, 33]. Because of its inherent communication-efficiency [33, 19] and its preservation of the privacy of sensitive local data [1, 43], it has gained substantial interest in the community [48, 35, 51], including studies on the convergence of the distributed system and the quality of the resulting model [30, 39, 50, 2]. While research in federated deep learning is usually focused on achieving a lower amount of communication, preserving the error rate, and/or preserving privacy [25, 48], quantifying uncertainty in the federated setup is not fully understood, yet.

In their work, Boughorbel et al. [3] propose using uncertainty, measured as generalization ability of the model, for weighted aggregation in the global model. Nevertheless, they do not concentrate on the uncertainty quantification techniques for federated learning.

In this work we extend the use of ensembles in federated deep learning as well as include two other popular approaches for uncertainty quantification: MC-dropout and stochastic weight averaging Gaussians (SWAG).

2 Preliminaries

In this section we give a brief overview about approaches to uncertainty quantification, as well as federated deep learning.

2.1 Uncertainty Quantification in Deep Learning

Let $D := \{(x_i, y_i)\}_{i \in [n]}$ be a dataset consisting out of n independent input-output tuples sampled from the same data generating distribution. A neural

network with parameters θ trained on D forms a model $p(Y|x^*, \theta)$ of the conditional probability distribution of the output Y given the input x^*. For making a prediction one is usually interested in finding the output y with the highest probability, i.e. $argmax_y p(Y = y|x^*, \theta)$[3]. One way to assess the uncertainty of such a prediction is to calculate the *Shannon entropy* of the predictive distribution. Let the output variable Y take values in a discrete set with K states, then the Shannon entropy is given by

$$H\left(p\left(Y|x^*, \theta\right)\right) = -\sum_{k=1}^{K} p\left(y_k|x^*, \theta\right) \cdot \log p\left(y_k|x^*, \theta\right) \quad . \tag{1}$$

The entropy reaches a minimal value of 0 iff one output value has probability 1 and the others probability 0, i.e. when the model has maximal certainty.

Bayesian models incorporate model uncertainty by taking the posterior distribution $p(\theta|D)$ of the parameters θ given the dataset D into account and estimating the expected prediction as

$$p(y|x^*, D) = \int p(y|x^*, \theta)p(\theta|D)d\theta \quad . \tag{2}$$

Since the posterior distribution for Bayesian neural networks is intractable, different approximation techniques have been proposed. In the following we will briefly introduce the approaches to uncertainty quantification that we have incorporated into a distributed framework in this paper.

Deep Ensembles. The most straightforward and simple approach for estimating uncertainty in neural networks is based on ensembles. The use of ensemble techniques in machine learning has been intensively studied. It is known as a way to improve the performance of weak classifiers in practice, not only for neural networks but also for other models, e.g. random forests [4]. Neural network ensembles were further leveraged by Lakshminarayanan et al. [28] to quantify the uncertainty of neural network predictions. They suggest to train a neural network S-times with different random initializations leading to S different models with parameter sets $\theta_1, \ldots, \theta_S$. For the final prediction all the predictions of every single model are averaged:

$$p(y|x^*) := \frac{1}{S}\sum_{s=1}^{S} p(y|x^*, \theta_s) \quad . \tag{3}$$

Next to assessing the uncertainty based on the entropy of $p(y|x^*)$ one can now also assess uncertainty by calculating the variance between network predictions, also referred to as *predictive variance*:

$$\sigma^2 = \frac{1}{S}\sum_{s=1}^{S} p\left(y|x^*, \theta_s\right)^2 - p\left(y|x^*\right)^2 \quad . \tag{4}$$

[3] For simplicity we will write $p(y|x^*, \theta)$ for $p(Y = y|x^*, \theta)$ in the following.

While surprisingly simple, uncertainty estimates from deep ensembles have shown to be competitive or even superior to the mathematically grounded uncertainty from Bayesian methods [28, 38]. Note, that while the implementation is straightforward, the computational burden is S-times as much as for training a single model.

Monte Carlo Dropout. Dropout is a regularization method first proposed by Hinton et al. [16] for reducing overfitting in deep learning by preventing complex co-adaptions of neurons [45]. The term *dropout* refers to the random deactivation of neurons of a neural network with probability α, called the droprate, during training time. When dropout is used for regularization it is only applied during training.

Gal and Ghahramani [10] showed, that if the network is trained with L2-regularization in addition and nodes are dropped also during inference (with the same probability as during training) the procedure will become an approximate Bayesian method. For deriving the final prediction, the probability of an output given a certain input is estimated multiple times for different sub-nets resulting from randomly dropping neurons and averaged output probabilities as in eq. (3) and the predictive variance can be estimated as in eq. (4), where θ_s now represents the parameters of the s-th sub-network.

Stochastic Weight Averaging Gaussian. Maddox et al. [32] propose another approximate Bayesian method, that exploits the trajectory of stochastic gradient decent (SGD). Their method is inspired by *Stochastic Weight Averaging* (SWA) [17], where starting from a pretrained solution, averaging the network parameters along the trajectory of SGD improves generalization. In SWAG normal distributions are placed over the parameters, where the mean of each parameter is calculated during training as in SWA by:

$$\theta_{\text{SWA}} = \frac{1}{T}\sum_{i=1}^{T}\theta_i \ , \tag{5}$$

where T is the number of SWA epochs. To efficiently calculate the covariance of the parameters, SWAG computes a running average of the second uncentered moment for each weight:

$$\overline{\theta^2} = \frac{1}{T}\sum_{i=1}^{T}\theta_i^2 \ . \tag{6}$$

After the last training step $\overline{\theta^2}$ and θ_{SWA} are combined to form a (in our case diagonal) covariance matrix over the parameters by

$$\Sigma_{diag} = diag(\overline{\theta^2} - \theta_{\text{SWA}}^2) \ . \tag{7}$$

During inference, the derived parameter distribution $\mathcal{N}(\theta_{\text{SWA}}, \Sigma_{diag})$ is treated as an approximate Bayesian posterior, equivalent to $p(\theta|D)$ in eq. (2). The integral is approximated by a Monte Carlo estimate, i.e. by averaging over samples

from the approximate posterior. In their original work Maddox et al. [32] experimentally showed that SWAG approximates the shape of the true posterior while being much less computationally expensive than traditional Bayesian methods.

2.2 Federated deep learning

In federated deep learning, the goal is to train a global neural network model f_{global} - which is a mapping from the input space to the output space - on m workers, each worker f_i with $i \in [m]$ holding a local dataset D_i drawn iid from the same data distribution. For that, each worker trains a local model with the same network structure as the global model and shares its model parameters θ_i with a coordinator. This coordinator averages the model parameters of local models and redistributes the averaged parameters

$$\overline{\theta} = \frac{1}{m} \sum_{i=1}^{m} \theta_i \tag{8}$$

so that the local workers continue training from $\overline{\theta}$. This process is iterated until a suitable stopping criterion is met.

The vanilla variant of federated learning averages all local models after a fixed number of training steps. The amount of communication spent on achieving a good performing global model is a critical characteristic of federated learning, since there exists a correlation: Investing more communication is expensive, but it leads to a better model. To reduce communication, random subsets of models can be averaged [33] or communication intervals can be adjusted dynamically [19]. To improve model quality, averaging can be replaced by other aggregation techniques, such as the geometric median [40] or the Radon point [21].

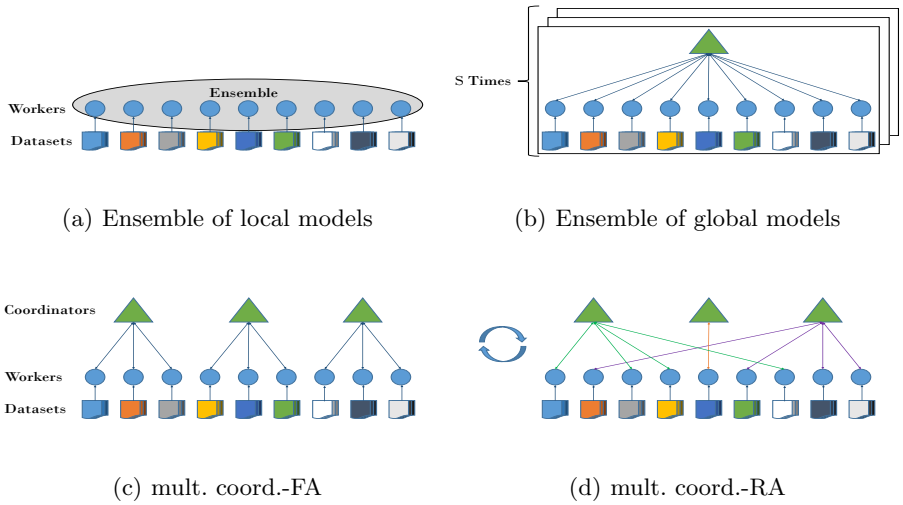
In our evaluation vanilla variant of federated learning serves as a baseline for the other approaches, since none of the special uncertainty quantification techniques are integrated there. We will refer to this approach as *global model*—as uncertainty is measured for the final global model at coordinator.

3 Leveraging uncertainty in federated deep learning

Given the preliminaries described in section 2, we now leverage the uncertainty methods into the federated learning setting.

3.1 Ensembles in federated deep learning

Different strategies to form an ensemble in a federated training scenario can be introduced, as illustrated in figure 3.1 and described in the following.



(a) Ensemble of local models

(b) Ensemble of global models

(c) mult. coord.-FA

(d) mult. coord.-RA

Fig. 3.1. Schema for different ways to build ensembles in a federated setup.

Ensemble of local models. A naive way to incorporate deep ensemble based uncertainty quantification into the federated setup is to consider the workers' local models as members of an ensemble. In order not to have the same model at every worker, one does not perform any communication with the coordinator, which leads to m separately trained models. These trained models f_i are used for the final prediction derived by averaging $f_i(x^*)$ and by replacing $p(y|x^*, \theta_s)$ in eq. (1) and eq. (4) by $f_i(x^*)$ for deriving the uncertainty measures. Note, that the idea of federated learning, where the local models benefit from others without seeing their data, is lost here.

Ensemble of global models. Here we describe another straightforward approach where the benefits of federated learning are kept, however, at the cost of a massive computational overhead. In *ensemble of global models* each worker trains S neural networks $f_{i,s}$, by using different random initialization to start with at each local model. For each of the S models we conduct the same procedure like in federated learning, increasing the computational effort by S times. However, because each run is independent of the others, this approach can easily be parallelized. It is still not practical in a real world setting with multiple computationally weak devices due to the increased computational effort and storage restrictions. The prediction for a new input x^* during the evaluation in each worker is given by the average over the S models, which results in

$$\hat{y} = \frac{1}{S} \sum_{s=1}^{S} f_{i,s}(x^*) \ . \tag{9}$$

Ensemble based on multiple coordinators. We also investigate ensemble strategies that are based on employing several coordinators in the federated training framework. In the first setup, we refer to as *fixed assignment (FA)*, the m workers are randomly grouped into A small subgroups, where each subgroup ($a \in [A]$) contains the same amount of workers. Each subgroup gets its own coordinator C^a and individually follows the federated learning process. After the last communication period the final predictions are derived by averaging the predictions $f(x^*; \bar{\theta}^a)$, $a \in [A]$ of each coordinator model.

Furthermore, we investigate an ensemble based on multiple coordinators with *random association (RA),* where each worker is randomly reassigned to one of the A coordinators C^a after each communication phase.

3.2 MC-Dropout in federated deep learning

We straightforwardly apply *federated MC-dropout* by transforming each local network f_i into a network where dropout with a droprate of $\alpha_i \in (0, 1)$ is applied during training and prediction, c.f. figure 3.2. The chosen droprate is the same for all workers, i.e. $\alpha_i = \lambda,\ \forall i \in [m]$. Note that this does not result in a change of the communication needed compared to the baseline. During inference each worker samples multiple subnets by randomly dropping neurons and the final prediction is derived by averaging the predictions of subnets in analogy to the centralized setting.

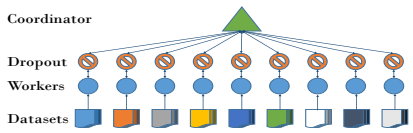
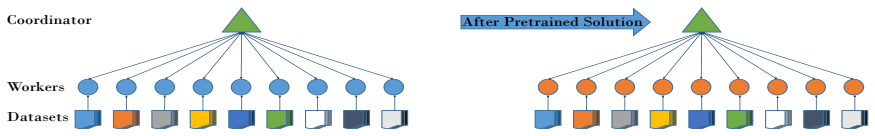


Fig. 3.2. Using dropout in a federated setup.

3.3 SWAG in federated deep learning

We conclude this section by leveraging SWAG to the federated setting as depicted in figure 3.3. *Federated SWAG* is implemented by first conducting vanilla federated training, but with one communication period less. This is followed by one period where each worker calculates the SWAG estimates of mean θ_{SWA} and variance Σ_{diag} as described in section 2. In the last communication period, each worker sends those two vectors (with a dimension each equal to the size of the network parameters θ) to the coordinator, which averages all mean values and all variance values and sends the averages back to the workers. Since both,

the aggregated θ_{SWA} and aggregated Σ_{diag}, are distributed back, the amount of communication in this period is doubled. For inference each worker conducts predictions based on draws from the estimated approximate posterior distribution $\mathcal{N}(\theta_{\text{SWA}}, \Sigma_{diag})$.



(a) Global model (b) Federated SWAG

Fig. 3.3. Baseline and federated SWAG.

4 Empirical Evaluation

We start with a description of the experimental setup. We conducted the experiments on two datasets, MNIST [49], which consists out of black and white images of size 28×28 of handwritten digits from zero to nine, and CIFAR-10 [27], which consists out of colored 32×32 images of ten different classes of common things like cars, cats, or airplanes. We used the standard training and test data split. For the experiments on MNIST we implemented a simulated distributed environment[4], where code was run on a single GPU, and did not take into account issues of real distributed systems, like race conditions during communication. The experiments on CIFAR-10 are conducted using the already existing DL-Platform [18], which enables federated deep learning on a large scale. For both datasets the hyper-parameters, especially the learning rate, were tuned to maximize validation set accuracy. The tuning was done separately for each experiment. An overview of the used experimental settings is given in appendix A. For the experiments on MNIST we used a simple fully connected network with two hidden layers with 128 neurons each and stochastic gradient decent as optimizer. Each experiment was run ten times with random initializations of the network parameters. For CIFAR-10 we used a ResNet18 architecture, and ran the experiment five times with different random initializations and using dropout for regularization [5]. For this, as well as for MC-dropout we applied a dropout-rate of

[4] Sourcecode available at: https://github.com/FloLins/Approaches-to-Uncertainty-Quantification-in-Federated-Deep-Learning

[5] Using dropout for regularization was necessary to reach a reasonable accuracy on CIFAR in our experiments. For comparison, results for the same setting without regularization can be found in the appendix in table 6.

$\lambda = 0.5$ for both datasets. For the federated setup we use 20 local learners and 1 coordinator. In the case of FA and RA we introduce 4 coordinators, analogously for the ensemble of global models we train 4 models at each local worker.

To investigate the ability of the different approaches to quantify uncertainty we measure the ability of the models to distinguish between (i) correctly classified and missclassified test examples and (ii) out of distribution data (OOD data) and the original in distribution data. To measure these abilities we first estimate the Shannon entropy given in eq. (1) and the predictive variance given in eq. (4) and then we calculate the *area under the receiver operating curve* (AUROC) [14] for classification based on this quantities. For calculating the AUROC for wrong and right predictions on the test dataset we sampled an equal amount of correctly and wrongly classified samples (to avoid class imbalances). We repeat this procedure three times for statistical accuracy. For models trained on MNIST we used the KMNIST [6] as OOD data, and for models trained on CIFAR-10—SVHN [37].

Because each setup differs in the amount of averaged predictions and computational complexity, we only compare them separately to the global model found by standard federated deep learning which serves as a baseline.

Table 1 Performance of ensemble approaches in comparison to vanilla federated training (denoted as global model).

Approach	Accuracy	Ent. AUROC	Var. AUROC
MNIST (10 runs) with KMNIST as out-of-distribution data			
Global model	97.70 ± 0.001	0.909 ±0.004	-
Ensemble of local models	94.13 ± 0.001	0.899 ±0.001	0.893 ±0.001
Ensemble of global models	**97.88** ± 0.001	**0.927** ±0.002	**0.931** ±0.002
Ensemble w. mult. coord.-FA	97.03 ± 0.001	0.916 ±0.002	0.914 ±0.003
Ensemble w. mult. coord.-RA	97.36 ± 0.004	0.906 ±0.002	0.878 ±0.006
MNIST (10 runs) uncertainty for wrongly classified data			
Global Model	97.70 ± 0.001	0.968 ±0.008	-
Ensemble of local models	94.13 ± 0.001	0.934 ±0.005	0.930 ±0.006
Ensemble of global models	**97.88** ± 0.001	**0.969** ±0.006	**0.966** ±0.006
Ensemble w. mult. coord.-FA	97.03 ± 0.001	0.959 ±0.007	0.950 ±0.007
Ensemble w. mult. coord.-RA	97.36 ± 0.004	0.962 ±0.006	0.945 ±0.011
CIFAR-10 on ResNet18 (5 runs) with dropout and SVHN as out-of-distribution data			
Global model	86.58 ± 0.289	0.924 ±0.011	-
Ensemble of local models	72.65 ± 0.251	0.679 ±0.023	0.689 ±0.033
Ensemble of global models	**89.00** ± 0.136	**0.937** ±0.009	**0.804** ±0.012
Ensemble w. mult. coord.-FA	83.43 ± 0.449	0.860 ±0.014	0.740 ±0.012
Ensemble w. mult. coord.-RA	86.72 ± 0.511	0.920 ±0.004	0.750 ±0.026
CIFAR-10 on ResNet18 (5 runs)with dropout, uncertainty for wrongly classified data			
Global model with dropout	86.58 ± 0.289	0.888 ±0.006	-
Ensemble of local models	72.65 ± 0.251	0.788 ±0.006	0.687 ±0.007
Ensemble of global models	**89.00** ± 0.136	**0.891** ±0.008	**0.868** ±0.006
Ensemble w. mult. coord.-FA	83.43 ± 0.449	0.858 ±0.005	0.815 ±0.007
Ensemble w. mult. coord.-RA	86.72 ± 0.511	0.887 ±0.005	0.861 ±0.009

Ensembles. We report the accuracy and the results for uncertainty based OOD detection and detection of wrongly classified data for the investigated ensemble methods in table 1. The *ensemble of local models* has a lower accuracy than the baseline on both datasets, which can be attributed to the fact that each local model only sees a limited amount of data. Even on the simpler MNIST dataset no satisfactory results are achieved.

Like in a centralized setting the *ensemble of global models*, where we used four global models, increases both accuracy and uncertainty quantification quality, but at the cost of an increased computational payload. This overhead could decrease its practical usability in the case that the training is performed on multiple computationally weak devices. In contrast, just increasing the amount of coordinators from one to four does not increase the computational complexity of the approach. A fixed association of workers to coordinators lead to less accurate results than a random association (compare Ensemble w. mult. coord.-FA to Ensemble w. mult. coord.-RA in table 1), which could be explained by the fact that each coordinator sees more data. We also experimented with randomly choosing the size of the random subgroups (results not shown), which however did not lead to significant differences in the results.

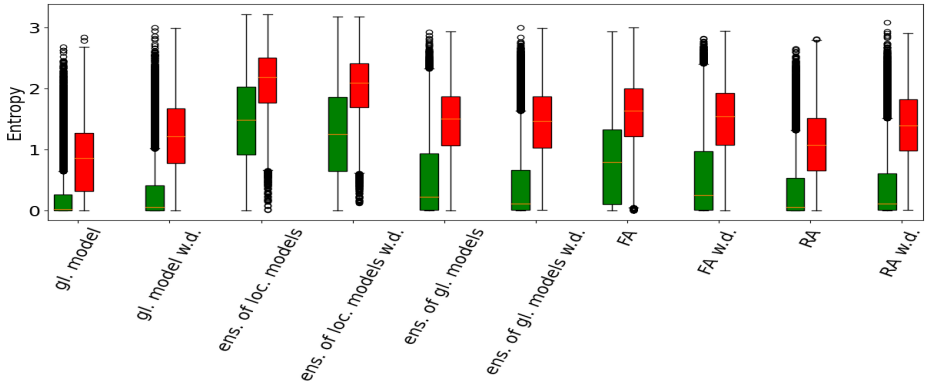


Fig. 4.1. The spread of entropy values for right (green) and wrong (red) predictions of CIFAR-10.

In figures 4.1, 4.2 we show the uncertainty for wrong and right predictions for the in distribution test dataset of CIFAR-10. One can see that generally the highest entropy values are produced by the ensemble of local models but entropy is high for correct and incorrect predictions. The other models, like *ensemble of global models*, show a significantly larger difference in entropy values for missclassified and correctly classified examples, which is reflected by the higher AUROC values.

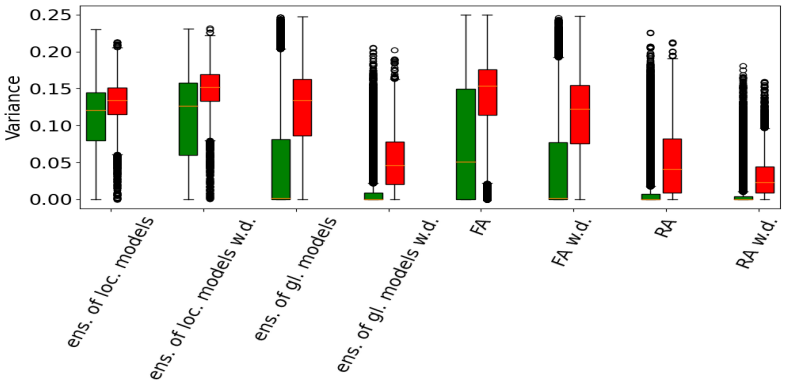


Fig. 4.2. The spread of variance values for right (green) and wrong (red) predictions of CIFAR-10.

MC-Dropout. The findings for applying dropout in the federated setting are twofold: First, in our experiments it increases the AUROC based on the entropy for both datasets when compared to the global model (c.f. table 2). This behavior relates to observations in the centralized setting, where MC-dropout is used as a simple and effective method for uncertainty quantification [10]. Second, for models trained on CIFAR-10, dropout also improves the test set accuracy when compared to the global model and is comparable with the accuracy derived from dropout as a regularization (global model with dropout). Because of the significant accuracy increase when using dropout during training, one possible direction for future work could be to investigate if the inherent prevention of co-adaptation while using dropout is beneficial during weight averaging.

SWAG. We conclude our experimental discussion with analyzing the effects of federated SWAG on test set accuracy and uncertainty. The results presented in table 3 demonstrate that SWAG produces a slightly higher test set accuracy compared to the global model on both datasets. Further, the AUROC as well as the box-plot in figure 4.3 shows that the ability to indicate erroneous predictions based on the entropy is also increased. Because SWAG can be applied to existing (already trained) models by using the current state as pre-trained solution the found improvements can easily be archived by applying one communication period of the SWAG algorithm, i.e. without a lot of computational overhead compared to vanialla ferderated learning. Variance based OOD detection as well as wrong and right prediction distinction (figure 4.4) is always inferior to using the entropy.

Table 2 Comparison of federated MC-dropout to global model approach.

Approach	Accuracy	Ent. AUROC	Var. AUROC
MNIST (10 runs) with KMNIST as out-of-distribution data			
Global model	**97.70** ± 0.001	0.909 ± 0.004	-
Federated MC-dropout	96.77 ± 0.001	**0.920** ± 0.002	**0.871** ± 0.003
MNIST (10 runs) uncertainty for wrong classified data			
Global Model	**97.70** ± 0.001	**0.968** ± 0.008	-
Federated MC-dropout	96.77 ± 0.001	0.948 ± 0.009	**0.914** ± 0.013
CIFAR-10 on ResNet18 (5 runs) with SVHN as out-of-distribution data			
Global model	77.88 ± 0.279	0.742 ± 0.019	-
Global model with dropout	**86.58** ± 0.289	**0.924** ± 0.011	-
Federated MC-dropout	86.32 ± 0.253	0.913 ± 0.010	**0.714** ± 0.007
CIFAR-10 on ResNet18 (5 runs) uncertainty for wrong classified data			
Global Model	77.88 ± 0.279	0.838 ± 0.006	-
Global model with dropout	**86.58** ± 0.289	**0.888** ± 0.006	-
Federated MC-dropout	86.32 ± 0.253	0.880 ± 0.007	**0.849** ± 0.008

Table 3 Comparison of federated SWAG to global model approach.

Approach	Accuracy	Ent. AUROC	Var. AUROC
MNIST (10 runs) with KMNIST as out-of-distribution data			
Global model	97.70 ± 0.001	0.909 ± 0.004	-
Federated SWAG	**98.16** ± 0.001	**0.918** ± 0.004	**0.893** ± 0.005
MNIST (10 runs) uncertainty for wrong classified data			
Global Model	97.70 ± 0.001	0.968 ± 0.008	-
Federated SWAG	**98.16** ± 0.001	**0.974** ± 0.006	**0.969** ± 0.009
CIFAR-10 on ResNet18 (5 runs) with dropout and SVHN as out-of-distribution data			
Global model	86.58 ± 0.289	0.924 ± 0.011	-
Fed. SWAG	**87.14** ± 0.170	**0.924** ± 0.009	**0.807** ± 0.025
CIFAR-10 on ResNet18 (5 runs) with dropout, uncertainty for wrong classified data			
Global model	86.58 ± 0.289	0.888 ± 0.006	-
Fed. SWAG	**87.14** ± 0.170	**0.892** ± 0.006	**0.856** ± 0.007

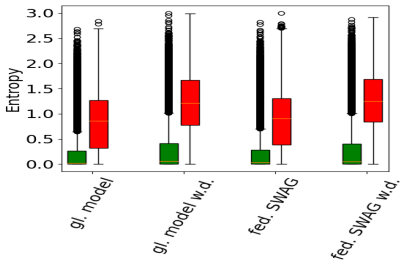


Fig. 4.3. The spread of entropy values for right (green) and wrong (red) predictions of CIFAR-10.

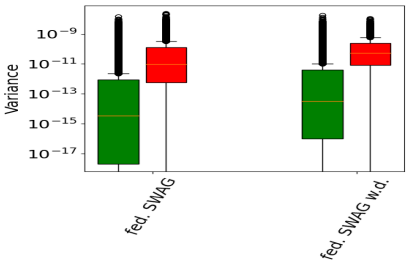


Fig. 4.4. The spread of variance values for right (green) and wrong (red) predictions of CIFAR-10.

5 Discussion and Conclusion

Due to the crucial relevance of uncertainty in trustworthy real world applications we investigated how approaches for uncertainty quantification can be applied in federated deep learning. More precisely we investigated ways to incorporate deep ensembles, SWAG and MC-dropout into the federated learning setup by changing network structure, communication protocols and training procedures. We believe, that the distributed setup provides further opportunities, as well as challenges for quantification of uncertainty and position our work as a baseline for possible future approaches.

The empirical results suggest that *ensembles of global models* and *federated SWAG* retain the model quality of standard federated learning while at the same time improve upon the the standard setting in terms of out-of-distribution (OOD) detection and detection of missclassified test data. The disadvantage of this approach is that it requires additional computation and storage for each global model that needs to be computed. One needs to evaluate this requirements for a real world application, since in applications with computational weak devices this approach could be infeasible.

Federated SWAG is less computational demanding, while still improving the prediction accuracy. In terms of prediction accuracy and uncertainty estimation abilities we found that the *federated SWAG* reaches values close to the *ensemble of global models*. Therefore we recommend the usage of *federated SWAG* due to the precise results and the high usability.

Further, the flexibility of the federated learning protocols allows multiple applications of uncertainty quantification in practice. *Ensembles based on multiple coordinators* could be beneficial for federated learning in moving objects, like mobiles or cars, because our empirical analysis shows that adding additional coordinators and assigning workers randomly to subgroups does not decrease both accuracy and capabilities for uncertainty quantification. Further no modification of the training algorithm nor network structure are needed to apply this method. Even adding uncertainty quantification methods to a running system is possible by using *federated SWAG*, since the actual state of model can be used as pre-trained solution.

Concluding, in this work we have empirically shown, that federated deep learning can benefit from approaches for uncertainty quantification in terms of accuracy and certainty of prediction while gaining additional design flexibility.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and the BMBF Project *Intrusion Detection in der Industrie 4.0 durch Fusion physikalischer Sensordaten mittels KI - mINDFUL*.

Bibliography

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318. ACM (2016)

[2] Adilova, L., Rosenzweig, J., Kamp, M.: Information-theoretic perspective of federated learning. In: Workshop on Information Theory and Machine Learning 2019. Association for Information Systems (2019)

[3] Boughorbel, S., Jarray, F., Venugopal, N., Moosa, S., Elhadi, H., Makhlouf, M.: Federated uncertainty-aware learning for distributed hospital ehr data. Machine Learning for Health (ML4H) Workshop at NeurIPS (2019)

[4] Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001), https://doi.org/10.1023/A:1010933404324

[5] Burdakis, S., Deligiannakis, A.: Detecting outliers in sensor networks using the geometric approach. In: Data Engineering (ICDE), 2012 IEEE 28th International Conference on. pp. 1108–1119. IEEE (2012)

[6] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical japanese literature. ArXiv abs/1812.01718 (2018)

[7] Deligiannakis, A., Kotidis, Y., Roussopoulos, N.: Processing approximate aggregate queries in wireless sensor networks. Information Systems 31(8), 770–792 (2006)

[8] Däubener, S., Fischer, A.: Investigating maximum likelihood based training of infinite mixtures for uncertainty quantification. In: Workshop on Uncertainty in Machine Learning ECML/PKDD (2020)

[9] Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting Adversarial Samples from Artifacts. arXiv preprint arXiv:1703.00410 (2017)

[10] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. pp. 1050–1059 (2016)

[11] Giatrakos, N., Deligiannakis, A., Garofalakis, M., Sharfman, I., Schuster, A.: Prediction-based geometric monitoring over distributed data streams. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. pp. 265–276. ACM (2012)

[12] Graves, A.: Practical variational inference for neural networks. In: Advances in Neural Information Processing Systems. vol. 24. Curran Associates, Inc. (2011), https://proceedings.neurips.cc/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf

[13] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning. pp. 1321–1330. Proceedings of Machine Learning Research, PMLR (06–11 Aug 2017)

[14] Hanley, J., Mcneil, B.: The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143 1, 29–36 (1982)

[15] Hinton, G.E., van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory. p. 5–13. COLT '93, Association for Computing Machinery, New York, NY, USA (1993), https://doi.org/10.1145/168304.168306

[16] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors (2012)
[17] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407 (2018)
[18] Kamp, M., Adilova, L.: Distributed Learning Platform (2020), https://github.com/fraunhofer-iais/dlplatform
[19] Kamp, M., Adilova, L., Sicking, J., Hüger, F., Schlicht, P., Wirtz, T., Wrobel, S.: Efficient decentralized deep learning by dynamic model averaging. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 393–409. Springer (2018)
[20] Kamp, M., Boley, M., Keren, D., Schuster, A., Sharfman, I.: Communication-efficient distributed online prediction by dynamic model synchronization. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 623–639. Springer (2014)
[21] Kamp, M., Boley, M., Missura, O., Gärtner, T.: Effective parallelisation for machine learning. In: Thirty-first Conference on Neural Information Processing Systems. pp. 6477–6488. Curran Associates (2017)
[22] Kamp, M., Bothe, S., Boley, M., Mock, M.: Communication-efficient distributed online learning with kernels. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 805–819. Springer (2016)
[23] Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? (2017)
[24] Keren, D., Sharfman, I., Schuster, A., Livne, A.: Shape sensitive geometric monitoring. IEEE Transactions on Knowledge and Data Engineering 24(8), 1520–1535 (2012)
[25] Konečný, J., McMahan, H.B., Yu, F.X., Richtarik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. In: NIPS Workshop on Private Multi-Party Machine Learning (2016)
[26] Kong, L., Sun, J., Zhang, C.: Sde-net: Equipping deep neural networks with uncertainty estimates (2020)
[27] Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research). Learning Multiple Layers of Features from Tiny Images (2009), http://www.cs.toronto.edu/~kriz/cifar.html
[28] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles (2016)
[29] Lechner, M., Hasani, R., Amini, A., Henzinger, T., Rus, D., Grosu, R.: Neural circuit policies enabling auditable autonomy. Nature Machine Intelligence 2, 642–652 (10 2020)
[30] Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581 (2019)
[31] MacKay, D.J.C.: A practical bayesian framework for backpropagation networks. Neural Comput. 4(3), 448–472 (May 1992), https://doi.org/10.1162/neco.1992.4.3.448
[32] Maddox, W.J., Izmailov, P., Garipov, T., Vetrov, D.P., Wilson, A.G.: A simple baseline for bayesian uncertainty in deep learning. In: Advances in Neural Information Processing Systems. pp. 13132–13143 (2019)
[33] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. pp. 1273–1282 (2017)

[34] Meijerink, L., Cinà, G., Tonutti, M.: Uncertainty estimation for classification and risk prediction on medical tabular data (2020)
[35] Mohri, M., Sivek, G., Suresh, A.T.: Agnostic federated learning. arXiv preprint arXiv:1902.00146 (2019)
[36] Neal, R.M.: Bayesian Learning for Neural Networks. Ph.D. thesis, University of Toronto, CAN (1995)
[37] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011)
[38] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J.V., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/f1ea154c843f7cf3677db7ce922a2d17-Paper.pdf
[39] Peterson, D., Kanani, P., Marathe, V.J.: Private federated learning with domain adaptation. arXiv preprint arXiv:1912.06733 (2019)
[40] Pillutla, K., Kakade, S.M., Harchaoui, Z.: Robust aggregation for federated learning. arXiv preprint arXiv:1912.13445 (2019)
[41] Ritter, H., Botev, A., Barber, D.: A scalable laplace approximation for neural networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Skdvd2xAZ
[42] Sharfman, I., Schuster, A., Keren, D.: A geometric approach to monitoring threshold functions over distributed data streams. ACM Transactions on Database Systems (TODS) 32(4), 23 (2007)
[43] Sharma, M., Hutchinson, M., Swaroop, S., Honkela, A., Turner, R.E.: Differentially private federated variational inference. arXiv preprint arXiv:1911.10563 (2019)
[44] Silva, S., Gutman, B.A., Romero, E., Thompson, P.M., Altmann, A., Lorenzi, M.: Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019). pp. 270–274. IEEE (2019)
[45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(56), 1929–1958 (2014), http://jmlr.org/papers/v15/srivastava14a.html
[46] Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. p. 681–688. ICML (2011)
[47] Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. Journal of Healthcare Informatics Research 5(1), 1–19 (2021)
[48] Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications (2019)
[49] Yann LeCun, Corinna Cortes, Christopher J.C.: The mnist database (visited on 2021-04-02), http://yann.lecun.com/exdb/mnist/
[50] Yao, X., Huang, T., Wu, C., Zhang, R., Sun, L.: Towards faster and better federated learning: A feature fusion approach. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 175–179. IEEE (2019)
[51] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. CoRR (2018)

A Appendix

The configuration for each experimental setup is summarized in table 4. Furthermore the optimized learning rate for each individual setup is presented in table 5.

Table 4 Hyperparameters used during the experiments.

Parameter	MNIST	CIFAR-10
Epochs per Communication Period	10	0.064
Batch Size	100	16
#Coordinators	4	4
#Workers	20	20
#Communication Periods	10	3125
OOD-Dataset	KMNIST	SVHN
Weight Decay	$1e-5$	$1e-4$
Repetitions for Ensemble of Global Models	4	4
Optimizer	SGD	SGD
Loss	Cross Entropy Loss	Cross Entropy Loss

Table 5 Learning rates used during the experiments.

Approach	MNIST	CIFAR-10
Global Model	0.1	0.01
Global Model w. dropout	-	0.1
Ensemble of local models	0.1	0.01
Ensemble of local models w. dropout	-	0.1
Ensemble of global models	0.1	0.01
Ensemble of global models w. dropout	-	0.1
Ensemble w. mult. coord.-FA	0.1	0.01
Ensemble w. mult. coord.-FA w. dropout	-	0.1
Ensemble w. mult. coord.-RA	0.1	0.01
Ensemble w. mult. coord.-RA w. dropout	-	0.1
Federated MC-dropout	0.05	0.1
Federated SWAG	0.1	0.01
Fed. SWAG w. dropout	-	0.1

Table 6 shows the results of CIFAR-10 when dropout was not applied as regularization method.

Table 6 Performance of CIFAR-10 without dropout as regularisation .

CIFAR-10 on ResNet18 (5 runs) with SVHN as out-of-distribution data			
Approach	Accuracy	Ent. AUROC	Var. AUROC
Global model	77.88 ± 0.279	0.742 ±0.019	-
Ensemble of local models	66.77 ± 0.564	0.468 ±0.019	0.648 ±0.031
Ensemble of global models	84.02 ± 0.218	0.776 ±0.02	0.719 ±0.015
Ensemble w. mult. coord.-FA	73.24 ± 0.791	0.584 ±0.023	0.631 ±0.019
Ensemble w. mult. coord.-RA	78.65 ± 0.291	0.760 ±0.028	0.704 ±0.024
Federated SWAG	78.72 ± 0.57	0.749 ±0.03	0.708 ±0.033
CIFAR-10 on ResNet18 (5 runs)uncertainty for wrong classified data			
Global Model	77.88 ± 0.279	0.838 ±0.006	-
Ensemble of local models	66.77 ± 0.564	0.760 ±0.004	0.630 ±0.006
Ensemble of global models	84.02 ± 0.218	0.862 ±0.004	0.823 ±0.008
Ensemble w. mult. coord.-FA	73.24 ± 0.791	0.808 ±0.004	0.742 ±0.012
Ensemble w. mult. coord.-RA	78.65 ± 0.291	0.847 ±0.006	0.826 ±0.006
Federated SWAG	78.72 ± 0.57	0.845 ±0.007	0.825 ±0.007