ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Hierarchical Variational Inference for Federated Learning

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Maksimov	Anton

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zurich, 09.10.2020

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Hierarchical Variational Inference for Federated Learning.

Student:	Anton Maksimov	Start Date:	20 Mar 2020
Advisors:	Luca Corinzia	End Date:	09 Oct 2020

Description

Federated learning is a recently emerged paradigm describing learning of a model using a server and a network of devices. These devices, called clients, retain a private dataset and transmit and receive information to/from the server in the form of agglomerated updates. This setting has some advantages with respect to train many models separately and on training a single model on pooled data, such as delegation of computations from server to devices, increasing of the effective amount of training data and possibilities to perform learning in a more secure and personalized way. These features solves many of the typical constraints found in applications, e.g. limited and heterogeneous storage and computational power of the devices, and impossibility of pooling the datasets in a centralized settings due to limited bandwidth and privacy concerns.

One of the recent approaches in federated learning is the use of the variational inference to handle a multi-task learning setting. It was shown to tackle non-IID data and perform multi-task learning of non-convex models successfully competing with the current state-of-the-art methods. However, to the best of our knowledge variational federated learning has been applied only to mean-field models. The goal of the project is to extend this approach to the hierarchical models by using shared hyperpriors on the parameters of the client models and thus inducing new dependencies between them. Hierarchical models can be especially useful when the data inherently comes from several distinct clusters of sources, e.g. language models with different local variations.

Related Work

The first known Federated Learning algorithm is *FedAvg* [1]. According to it a server performs averaging of client models, which are locally updated using stochastic gradient descent on private data. However, it was shown already in the original paper that FedAvg is not robust in case of non IID-data. One way to overcome this is to introduce proximal term which pushes local models not to deviate too much during one local updating round [2], though this modification still doesn't lead to consistently better than FedAvg learning over non–IID data [2].

Multi-task learning (MTL) aims at tackling this problem, having as a goal learning models for multiple related tasks simultaneously. It can be classified depending on how relationships between tasks are introduced: if their structure is known *a priori* or it should be inferenced from the data [3] (hard and soft parameter sharing correspondingly), if we would like to improve the performance of all the tasks simultaneously or only for some subset of them (symmetric and asymmetric MTL) [4].

Distributed MTL aims at learning from data distributed over a network. In a federated setting the structure between models fitted to local datasets frequently exists and this fact makes it possible to improve the learning [3, 5]. Nevertheless, the approach proposed in [3] allows to handle only convex losses, which are not applicable to deep learning models. In order to overpass this limitation one can use *variational inference* [6]. This technique makes it possible to efficiently learn over highly non–IID data distributions.

One of the improvements of this method could be the expanding the star–shaped Bayesian network of parameters to a hierarchical one [7], where distributions of the modes parameters are defined by the other hyper-priors distribution which are learned during the optimization. The learning method should allow flexible generalization of common network knowledge and specialization of nodes according to their local data [8]. Different variational inference methods, such as black-box variational inference [9] or normalizing flows [10] could be used.

Note, that the notion of *hierarchical federated learning* can be used in a sense of introducing additional layer of auxiliary servers which first collect data from their children clients in order to reduce communication cost (e. g. structure server — cell base stations — connected to them mobile users [11, 12]), but this type of hierarchy differs from considered in this work, as it assumes that all clients have common data generating process and hierarchy comes from the construction of the communication network, whereas we consider clients belong to several different groups with their own data generating processes, which means hierarchy of their latent parameters.

Table 1: Initial plan

Nr.	Work Package
1	Review literature
2	Implement hierarchical Bayesian network on the MNIST data splitted according to digits similarity
3	Extend approach to other FL datasets (FEMNIST, VSN, HAR, Shakespeare etc.)
4	Implement variation of Virtual algorithm with additional shared hyper-prior on simple datasets
5	Outline of written thesis
6	Draft of written report
7	Hand-In written report

Theory derivation for hierarchical federated update

Having K clients, θ are server parameters (network weights $(\boldsymbol{\mu}^s, \boldsymbol{\sigma}^s)$, ϕ_i are the corresponding client parameters $(\boldsymbol{\mu}_i^c, \boldsymbol{\sigma}_i^c)$, and corresponding to server and clients λ_i^s and λ_i^c (s. t. $\lambda^s = \{\lambda_i^s\}_{i=1}^K, \lambda^c = \{\lambda_i^c\}_{i=1}^K$) are the group assignment distributions with G groups (simplexes in $\mathbb{R}^G$, which we assume to have uniform prior distributions there).

We have a mean–field proxy posterior distribution:

$$q(\theta, \phi, \lambda) = s(\theta, \lambda^s)\, c(\phi, \lambda^c) = \left(\prod_{i=1}^K s_i\left(\theta, \lambda^s\right)\right) \left(\prod_{i=1}^K c_i\left(\phi_i, \lambda_i^c\right)\right)$$

where we are considering every distribution to be a Gaussian mixture of G groups

$$\begin{cases} s_i\left(\boldsymbol{\theta}, \lambda_i^s\right) = \sum_{j=1}^G \lambda_i^s(j)\, \mathcal{N}\left(\boldsymbol{\mu}^s(j), \boldsymbol{\sigma}^s(j)\right) \\ c_i\left(\phi_i, \lambda_i^c\right) = \sum_{j=1}^G \lambda_i^c(j)\, \mathcal{N}\left(\boldsymbol{\mu}^c(j), \boldsymbol{\sigma}^c(j)\right) \end{cases}$$

Following the idea of [6], we then rewrite KL–divergence and get updating rule for weights

$$D_{KL}\left(s^{(t)}(\boldsymbol{\theta},\lambda^s)\,c^{(t)}(\phi,\lambda^c)\middle|\middle|\frac{s^{(t)}(\theta,\lambda^s)\,c^{(t)}(\phi,\lambda^c)}{s_i(\theta,\lambda^s)\,c_i(\phi_i,\lambda_i^c)}\,p(\theta,\phi_i,\lambda^s,\lambda_i^c|\mathcal{D}_i)\right)$$

$$=\int d\boldsymbol{\lambda}^s\int d\boldsymbol{\theta}\,s^{(t)}(\boldsymbol{\theta},\lambda^s)\,\log s_i(\boldsymbol{\theta},\lambda_i^s)\,\underbrace{\int d\boldsymbol{\phi}\,d\boldsymbol{\lambda}^c\,c^{(t)}(\boldsymbol{\phi},\lambda^c)}_{=1}+\int d\boldsymbol{\phi}\,d\boldsymbol{\lambda^c}\,c^{(t)}(\boldsymbol{\phi},\lambda^c)\log c_i\left(\boldsymbol{\phi}_i,\lambda_i^c\right)\underbrace{\int d\boldsymbol{\theta}\,d\boldsymbol{\lambda^s}\,s^{(t)}(\boldsymbol{\theta},\lambda^s)}_{=1}-$$

$$-\int d\boldsymbol{\lambda^s}\int d\boldsymbol{\lambda^c}\int d\boldsymbol{\theta}\int d\boldsymbol{\phi}\,s^{(t)}(\boldsymbol{\theta},\lambda^s)\,c^{(t)}(\boldsymbol{\phi},\lambda^c)\log p\left(\boldsymbol{\theta},\boldsymbol{\phi}_i,\lambda^s,\lambda_i^c\mid\mathcal{D}_i\right)$$

$$=\int d\boldsymbol{\lambda}^s\int d\boldsymbol{\theta}\quad\underbrace{s^{(t)}(\boldsymbol{\theta},\lambda^s)}_{s_i(\boldsymbol{\theta},\lambda_i^s)\frac{s^{(t-1)}(\boldsymbol{\theta},\lambda^s)}{s_i^{(t-1)}(\boldsymbol{\theta},\lambda_i^s)}}\quad\log\frac{s_i(\boldsymbol{\theta},\lambda_i^s)\,s^{(t)}(\boldsymbol{\theta},\lambda^s)}{s^{(t)}(\boldsymbol{\theta},\lambda^s)\,p(\boldsymbol{\theta})}+\int d\boldsymbol{\lambda}_i^c\int d\boldsymbol{\phi}_i c_i^{(t)}\left(\boldsymbol{\phi}_i,\lambda_i^c\right)\log\frac{c_i^{(t)}(\boldsymbol{\phi}_i,\lambda_i^c)}{p(\boldsymbol{\phi}_i)}-$$

$$-\int d\boldsymbol{\lambda}^s\int d\boldsymbol{\lambda}_i^c\int d\boldsymbol{\theta}\,s^{(t)}(\boldsymbol{\theta},\lambda^s)\int d\boldsymbol{\phi}_i\,c_i^{(t)}\left(\boldsymbol{\phi}_i,\lambda_i^c\right)\left(\log p\left(\mathcal{D}_i\mid\boldsymbol{\theta},\boldsymbol{\phi}_i,\lambda^s,\lambda_i^c\right)-\underbrace{\log p\left(\mathcal{D}_i\right)}_{\text{const}}-\underbrace{\log\frac{1}{p(\lambda^s)}}_{\text{const}}-\underbrace{\log\frac{1}{p(\lambda_i^c)}}_{\text{const}}\right)$$

using normalisation of pdfs and Bayesian rule $p\left(\boldsymbol{\theta},\boldsymbol{\phi}_i,\lambda^s,\lambda_i^c\mid\mathcal{D}_i\right)=\frac{p(\mathcal{D}_i|\boldsymbol{\theta},\boldsymbol{\phi}_i,\lambda^s,\lambda_i^c)p(\boldsymbol{\theta})p(\boldsymbol{\phi}_i)p(\lambda^s)p(\lambda_i^c)}{p(\mathcal{D}_i)}$. Constants in the end (because of initial uniform priors) lead to the constant shift of the loss, therefore not influencing the result, and could be dropped. Therefore, using the fact that $\frac{s_i(\boldsymbol{\theta},\lambda_i^s)}{s^{(t)}(\boldsymbol{\theta},\lambda^s)}=\frac{s_i^{(t-1)}(\boldsymbol{\theta},\lambda_i^s)}{s^{(t-1)}(\boldsymbol{\theta},\lambda^s)}$ we obtain the loss of update

$$\mathcal{L}(s_i(\boldsymbol{\theta},\lambda^s),c_i(\boldsymbol{\phi}_i,\lambda^s)=D_{KL}\left(s_i(\boldsymbol{\theta},\lambda_i^s)\frac{s^{(t-1)}(\boldsymbol{\theta},\lambda^s)}{s_i^{(t-1)}(\boldsymbol{\theta},\lambda_i^s)}\middle|\middle|p(\boldsymbol{\theta})\frac{s^{(t-1)}(\boldsymbol{\theta},\lambda^s)}{s_i^{(t-1)}(\boldsymbol{\theta},\lambda_i^s)}\right)+D_{KL}\left(c_i^{(t)}\left(\boldsymbol{\phi}_i,\lambda_i^c\right)\middle|\middle|p\left(\boldsymbol{\phi}_i\right)\right)-$$

$$\mathbb{E}_{\substack{s^{(t)}(\boldsymbol{\theta},\lambda^s)\\ c_i^{(t)}(\boldsymbol{\phi}_i,\lambda_i^c)}}\log p\left(\mathcal{D}_i\mid\boldsymbol{\theta},\boldsymbol{\phi}_i,\lambda^s,\lambda_i^c\right)$$

which is very similar to the non-hierarchical one in [6].

Practical part

Model description

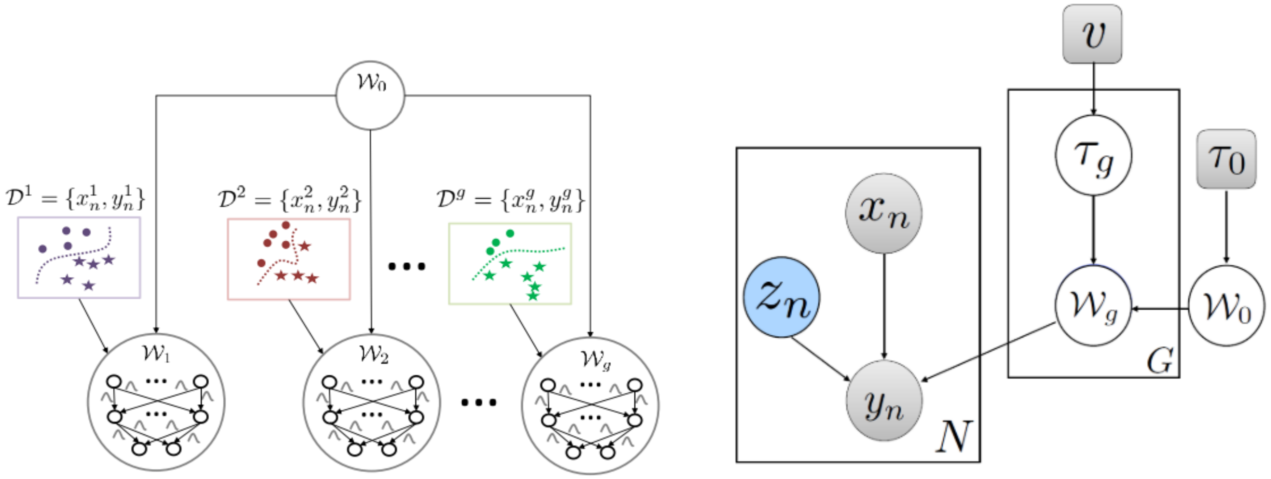
First, we implement the non-federated hierarchical Bayesian neural network following the paper on gesture classification [13], for which we were not able to find the code. For the structure description see the fig. 1.

We don't use layers of *tensorflow-probability* and produce code almost from scratch being inspired by the code from the Krasser's blog [14] as an illustration to the paper [15], using *TensorFlow Functional API*. We adapt it to the MNIST dataset with several groups represented by handwritten digits which are somehow similar from human (ours) point of view (e.g. 1 is similar to 7, 2 – to 5 etc. We use 3 such a groups). For the TensorFlow model structure see the fig. 2.

The loss consists of estimated by sampling ELBO and added to it cross-entropy between the layer output and the true label. Overall weight of ELBO is the same as the cross-entropy loss and is scaled accordingly to the number of groups in the hierarchy.

Experiments

The code is available on github. Data from MNIST is shuffled and divided as 60000 digits for training and 10000 for testing.



(a) Weights sets (”zero” and ”group-wise”) and datasets of the network

(b) Graphical representation of the model. Group-wise weights (their mean and variance as assumed normal distributions) $\mathcal{W}_g$ have learnable prior τ_g relative to ”zero”-weights $\mathcal{W}_0$. They have their own ”wide” priors v and τ_0. Group membership z_n (exactly known or deduced) affects the final class y_n, which is found using it, the feature x_n and $\mathcal{W}_g$.

Figure 1: Structure of the hierarchical Bayesian non-federated model, pictures are taken from [13].

As the main class-inference model we use single dense variational layer with no activation (surprisingly, Softmax or ReLU activation don’t allow to train the model effectively, as, probably, the output during training is, according to [13], the weighted sum of group-wise calls on the data, and non-linear activation somehow destroys this additive nature of the model).

For Monte-Carlo sampling in order to estimate the resulting class membership probabilities we use only 5 samples, as with bigger amount, as we noticed, initial learning is more spurious, see fig. 3 (because probably increased amount of samples which are too far from the mean makes the loss blow-up at some steps).

The ”group-inference” LeNet model is trained for 100 epochs till saturation.

Other the best used parameters: $\tau_0^{-1} = 1, v = 0.3,$ learning rate $= 0.1$, the same parameters are near-the-best for non-hierarchical training (except v, which it doesn’t have).

Results

Resulting test accuracy after 100 epochs much bigger (table 2) than the one of the model with non-hierarchical variational inference (VI) with the best found prior scale. It is also better than when using simple dense layer.

For dynamics of accuracy on train and test data for hierarchical and non-hierarchical variational models, see fig. 4

To test whether the weights corresponding to the groups of digits really train specifically, we plot on the fig. 5 the accuracy of prediction depending on the true label.

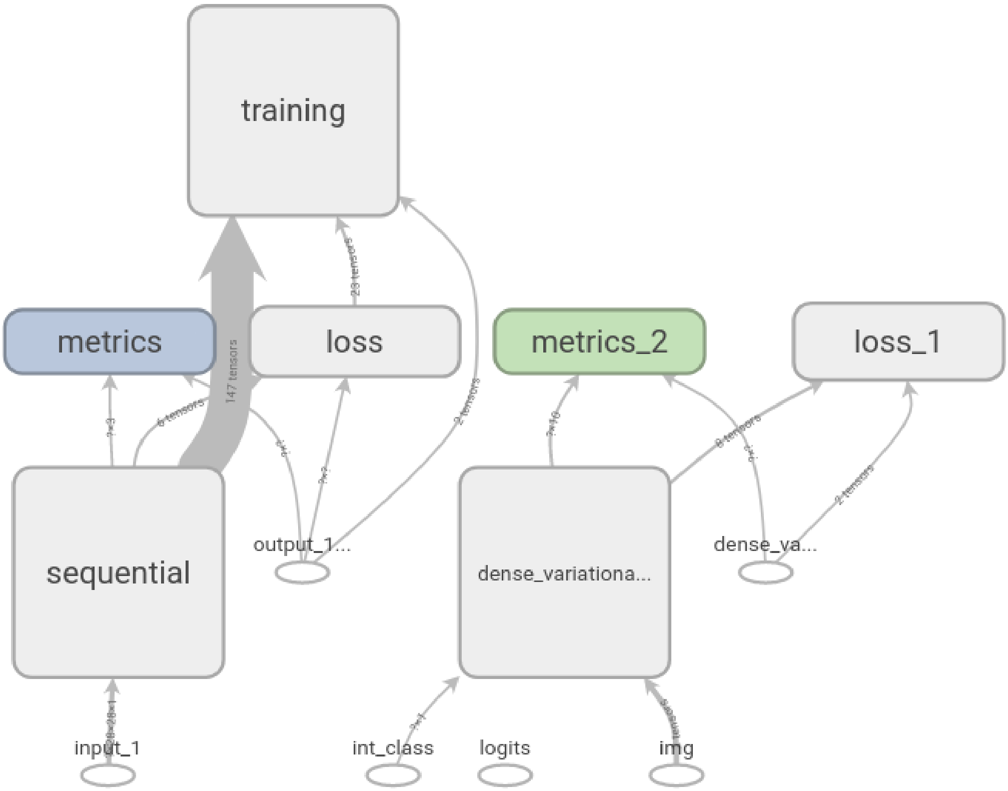
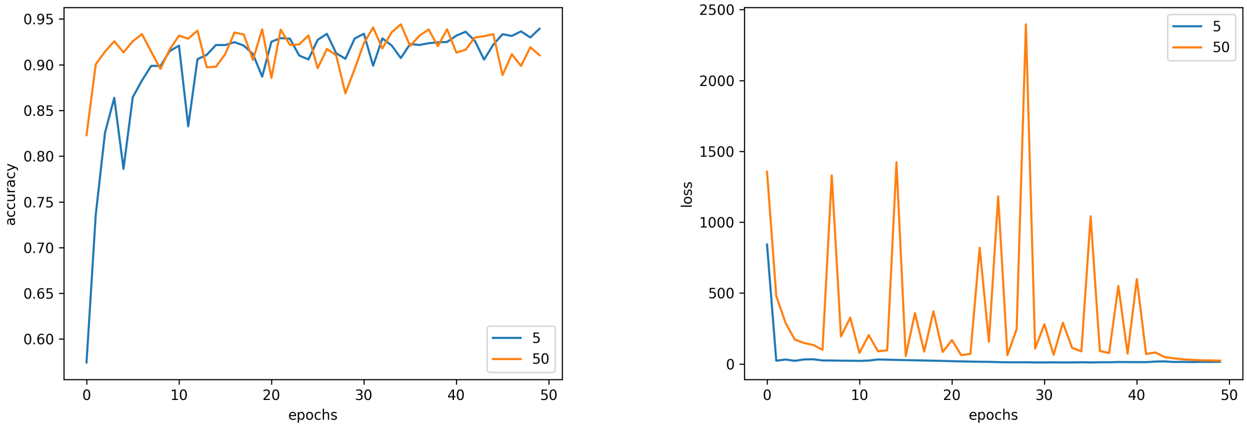


Figure 2: Full model structure. The left part (LeNet [16] structure) predicts probabilities of group-membership (we used 3 groups), then its output is used in the right part (dense variational grouped layer) in "predicting" mode, where it uses logits from the left part instead of integer exact memberships, whereas exact ones are used in the model's "training" mode.

Type	Accuracy
Dense - softmax	0.916 ± 0.003
Variational non-hierarchical	0.79 ± 0.02
Variational hieararchical	0.937 ± 0.003

Table 2: Comparison of methods' test accuracy after 100 epochs on single layer, each run is done for 5 repetitions with shuffled data



(a) Test accuracy

(b) Test loss

Figure 3: Difference when changing sampling number in hierarchical model.

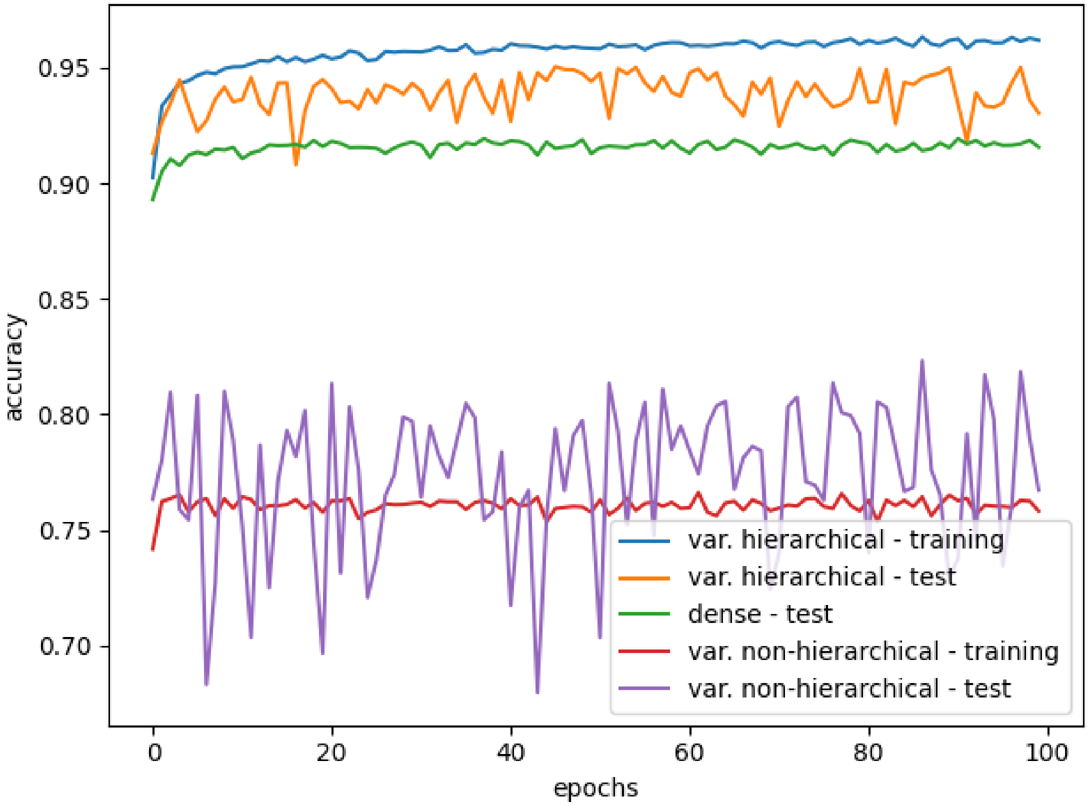
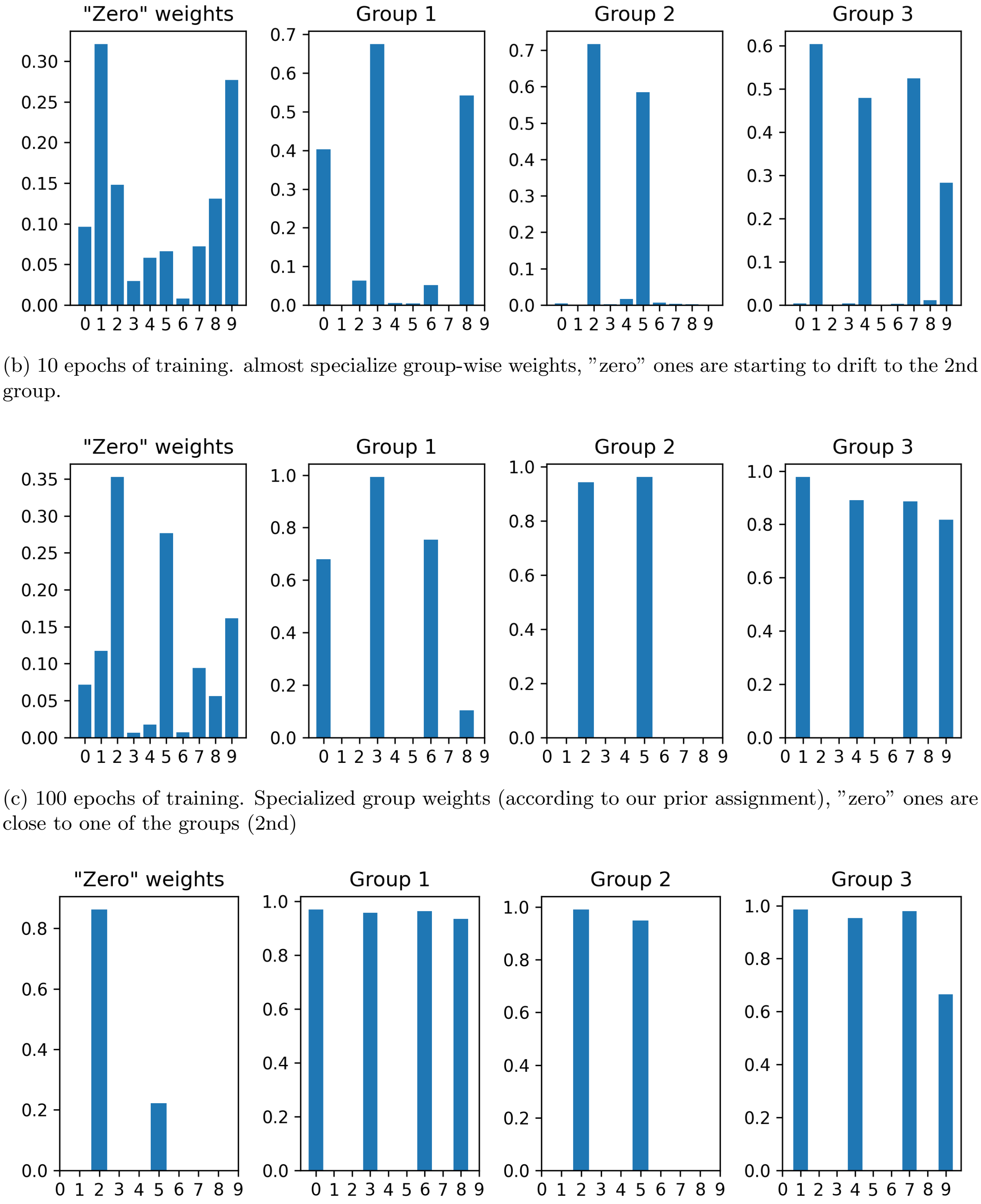


Figure 4: Training and testing accuracy during training for different methods. Variational methods have bigger variance of result, which could potentially be tackled by reducing learning step and increasing sample number in late training stages, or implementing Local Reparameterization [17] instead of usual non-centered reparameterization. Plus hierarchical method improves over non-hierarchical and has lesser variance too.

Figure 5: Accuracy of prediction depending on the true label for different types of weights and throughout training procedure.

(a) 1 epoch of training. Not yet specialized weights, especially ”zero” ones.



(b) 10 epochs of training. almost specialize group-wise weights, ”zero” ones are starting to drift to the 2nd group.

(c) 100 epochs of training. Specialized group weights (according to our prior assignment), ”zero” ones are close to one of the groups (2nd)

Resume

- Hierarchical variational model improves in terms of accuracy over simple dense and non-hierarchical variational ones when tested on MNIST dataset.
- The variance of hierarchical model is also less than of non-hierarchical one.

Future research

- Modification of the model for other types of data.
- Implementation of the federated type parameters' updates.

Project Specific Expectations

These goals express the expectations w.r.t which the work will be evaluated. This list should provide a reference, but not a simple checklist - independent initiative by the student which leads to results not listed here is encouraged.

Minimum Expectations

Implementation of the hierarchical Bayesian neural network [13] and its testing.

Expectations

Development of the algorithm for hierarchical variational federated learning using shared hyper-prior on the Virtual original algorithm. Its practical implementation and testing on real–world data. Several variational inference methods and datasets used.

Above Expectations

Exploration of new ideas in the area of hierarchical latent variational model in the area of F-MTL.

General Expectations

The following expectations are based on the evaluation sheet used for this thesis[1]. They should provide a general orientation about the expected workstyle and conduct.

Minimum Expectations

The student acquires a solid working knowledge of the relevant concepts and is therefore able to participate in discussions and implement the thesis. All work performed is well structured and documented as to enable efficient follow-up work. In addition to continuous documentation, this is particularly true for the final written thesis. All tasks which are agreed upon by supervisor and student are implemented. The expected quality of results corresponds to what the student can possibly achieve in the absence of major unexpected problems. The main contents of the thesis are presented in an informative but concise talk to an expert audience.

Expectations

The student acquires a good working knowledge of the relevant concepts and is therefore able to contribute in discussions and efficiently implement the thesis. Documentation of the performed work uses all good practice tools which are available for this purpose. The final report reflects the student's good knowledge about the relevant topics and enables detailed comprehension the thesis' contributions. In addition to the agreed tasks, the student regularly extends his research to questions which arise naturally during the work, but have not been discussed explicitly. The quality of the results corresponds to what is feasible in the context of a student thesis. The final presentation includes careful processing and structuring of the contents in order to effectively and illustratively convey information to the audience.

[1] https://www.ethz.ch/content/dam/ethz/special-interest/itet/department/Studies/Forms/SAMA_Bewertungsblatt_EN.pdf

Above Expectations

The student independently acquires knowledge which enables new perspectives and provides innovative suggestions. Documentation follows standards without compromises and shows a high commitment to detail while being efficiently useable by others. This applies especially to the use for a potential publication. There is a regular and significant attempt to explore all implications of the discussed tasks and make extensions and beneficial modifications to the original questions. The quality of the results is as good as the problem permits, major problems are solved if possible and indicate a high student performance. The final presentation delivers an impressive account of the thesis. The preparation and structure enables a seamless information flow. Great care is recognizable in the illustrations and graphics. The audience is captured and challenged.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 1273–1282, PMLR, 20–22 Apr 2017.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems 2020*, pp. 429–450, 2020.

[3] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 4424–4434, Curran Associates, Inc., 2017.

[4] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, (Arlington, Virginia, USA), p. 733–742, AUAI Press, 2010.

[5] A. R. Gonçalves, F. J. V. Zuben, and A. Banerjee, "Multi-task sparse structure learning with gaussian copula models," *Journal of Machine Learning Research*, vol. 17, no. 33, pp. 1–30, 2016.

[6] L. Corinzia and J. M. Buhmann, "Variational federated multi-task learning," 2019.

[7] R. Ranganath, D. Tran, and D. M. Blei, "Hierarchical variational models," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 2568–2577, JMLR.org, 2016.

[8] M. N. H. Nguyen, S. R. Pandey, K. Thar, N. H. Tran, M. Chen, W. Saad, and C. S. Hong, "Distributed and democratized learning: Philosophy and research challenges," 2020.

[9] R. Ranganath, S. Gerrish, and D. Blei, "Black Box Variational Inference," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* (S. Kaski and J. Corander, eds.), vol. 33 of *Proceedings of Machine Learning Research*, (Reykjavik, Iceland), pp. 814–822, PMLR, 22–25 Apr 2014.

[10] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1530–1538, PMLR, 07–09 Jul 2015.

[11] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Edge-assisted hierarchical federated learning with non-iid data," *ArXiv*, vol. abs/1905.06641, 2019.

[12] M. S. H. Abad, E. Ozfatura, D. GUndUz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8866–8870, 2020.

[13] A. Joshi, S. Ghosh, M. Betke, S. Sclaroff, and H. Pfister, "Personalizing gesture recognition using hierarchical bayesian neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 455–464, July 2017.

[14] "Martin Krasser's blog." http://krasserm.github.io/2019/03/14/bayesian-neural-networks/.

[15] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2575–2583, Curran Associates, Inc., 2015.