Decentralized and Heterogeneous Collaborative Learning for Personalized On-Device Healthcare Analytics

Sheng Xu[1]

Abstract

The topic of E-health has grown its popularity over the past few years due to the increasing awareness of people to their health as well as the advancement of wearable personal devices. It has been widely demonstrated that machine learning algorithms could detect human diseases at an early stage accurately and efficiently. However, these applications usually require aggregation of either data or gradients, which inevitably raises the issue of privacy protection and timeliness. In addition, machine learning models trained on global data could be biased when applied to individuals, thus resulting in declined overall model effectiveness. In this study, we propose a decentralized collaborative learning framework using knowledge distillation and adaptive weighted loss control method to optimize the model performance over edge devices. We avoid the share of data or gradients to achieve higher level of accuracy for edge devices compared to existing methods. To address the issue of quantification of neighbor devices' influence, we propose adaptive weighted knowledge distillation loss control. Light-weighted deployment and standard IoT communication protocols are involved to simulate real-world scenarios. We conduct the experiment on two real-world datasets which reveal the validity and improvement of our proposed methods for fully decentralized learning.

Keywords: E-Health, Decentralized Machine Learning

Introduction

E-Health technologies have hugely reshaped the industry of personal health care, including real-time health monitoring using sensors built in wearable devices. For instance, the latest apple watch has the capability to detect one's irregular ECG signal instantaneously, which could help know the potential heart disease in advance and avoid the risk of sudden heart attacks. It could

play a critical role in medical scenarios as it can help people realize their potential physical issues and take action earlier. It also improves doctor's diagnosis as it records detailed data logs.

However, most of the state of the art models require central servers to aggregate information such as raw data and labels to enable training. These approaches put a challenge on nowadays stricter privacy protection regulations. Centralized learning will inevitably cause privacy issues as the data could leak in many stages of aggregation. Besides, the gathering of all users' data is no longer acceptable under certain regulations. This creates a dilemma for those who want to apply the data for positive usage.

Federated learning[1] has been developed to tackle this issue. The model is duplicated to all devices and each device model will train on its local data, share the gradients with central server, and update parameters using the processed results aggregated by the central server. This seems a more safe way in terms of privacy preserving. Nevertheless, this approach cannot alleviate some shortcomings that traditional central training has. One critical problem is that federated learning requires identical model structure across devices to share gradients and parameters which brings about the computing capacity issue. For personal devices, they are often of limited and different memory sizes, computing powers, and etc. The designed model must fit the device with the least memory and computing power in order to enable the collaboration, which will inevitably reduce the precision of overall model performance. Thus federated learning is more fit to computing clusters across organizations such as server clusters across hospitals or research institutions.

More recently, Bistritz et al.(2020)[2] proposed distributed distillation to avoid the limitations posed by federated learning. Their method achieved decentralized training of machine learning models across edge devices using a common public, unlabeled dataset to perform cross device knowledge distillation. The result showed same of higher overall performance compared to FedAvg, however, their results would be significantly impacted by how well the common dataset is constructed. That is to say, the model would be more likely to have better performance if the device data distribution is similar to the reference dataset which is true in their case as they used randomly sampled image data to conduct their experiments. Nevertheless, under real world scenarios, a large proportion of the data are none independent and identically distributed, which brings about the challenges of training them in the fully decentralized environment.

We establish heterogeneous model training framework to leverage real-world light-weighted IoT devices' power to perform predictive collaborative training on their local data. To address the problem of collaborative training across heterogeneous model structures, we learn from the idea of federated knowledge distillation[3][4] and improve it with adaptive weighted loss control to perform neighbor device selection and determine the amount of knowledge that should be transferred among the group of edge devices. We achieve same or higher overall performance on real-world datasets compared to some of the

existed approaches. In addition, we use cross-platform compatible deployment and standard light-weight IoT communication protocols, which could be easily applied to enormous amount of existed IoT devices to leverage its power for E-Health diagnosis and other applications.

Results

Overall framework

To begin, we introduce the overall framework of DecentKD for collaboration between devices with heterogeneous architectures (Fig. 1). The framework enables devices learn personalized models collaboratively and preserve privacy at the same time. The DecentKD environment is fully decentralized and does not require any central server to help establish communication or data transfer. All devices are anonymous to each other, no raw data or model gradients would be shared. The edge device collects its user's data locally, and generates its local model based on its computing capability. The local device model is decoupled into two parts: initial simple residual unit as feature extractor and subsequent deep units as local learning blocks for processing local and neighbour feature maps. The information encoded by the personalized local feature extractor will be broadcast to its neighbours periodically, and each neighbour will send the calculated soft labels back to the original owner using its local sub model. The edge device aggregate all valid soft labels at the moment and use one step stitch method to choose collaborators and perform knowledge distillation to transfer useful information from neighbours. The interaction is started after the model converges on its local data and executed asynchronously over multiple iterations until the local model achieves significant level of accuracy. Overall, the encoded high dimensional feature maps and soft label based knowledge distillation avoid the direct share of raw data and labels, thus achieve the privacy preserved heterogeneous model collaboration in a fully decentralized environment.

Performance evaluation

Our experiments are conducted on two tasks using corresponding real world e-health dataset. The first task is sleep apnea detection which needs to predict whether a patient have sleep breathing disorder based on his ECG time-series data. We use the Physionet Apnea-ECG dataset[5] for this task. It consists of 70 records of continuous digitized ECG signal from both apnea patients and normal health people. Each record has a set of binary annotations derived by human experts to identify the apnea status of every minute. The sample rate of ECG signal is 100 per second, which means that there will be an annotation for every 6,000 data points. We perform sample level normalization to minimize the influence of data scale differences. Each recording is treated as one local dataset for an edge device, meaning that we will have 70 simulated edge devices to perform decentralized training. The second task is

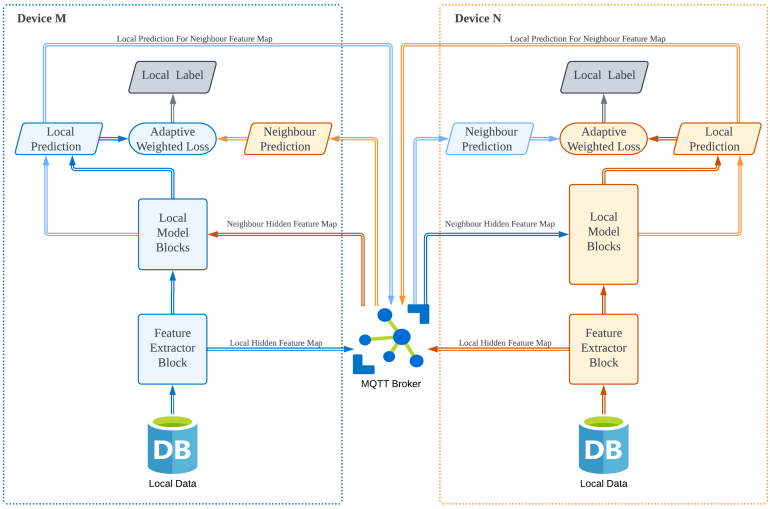

Fig. 1 The overall framework of DecentKD. Each user device generates a heterogeneous local model that fits its computation capability and learns a personalized model based on locally collected data. Each device model is decoupled into two parts: feature extractor block and local model blocks. All feature extractor blocks' output dimension across devices are identical to enable intermediate information exchange and sub model forward computation in a privacy preserved way. The devices share their encoded feature maps to neighbours and use returned soft predictions computed by neighbour sub models to perform adaptive neighbour selection and weighted knowledge distillation. An optional message queue broker is used to help optimize the data flow for IoT devices to establish communication and transmit information.

In our experiments, we refer to standard 1D ResNet[6] to design our heterogeneous device model structure by actively change the number of residual units in the network. We sample the number of blocks a device should have based on its local dataset size. The overall device model complexity ranges from ResNet-8 to ResNet-32. We compare DecentKD with several groups of baseline models. The first baseline group consists (1) Centralized ResNet-8, (2) Centralized ResNet-32, which correspond to the simplest model structure, and the most complex model structure, respectively. The centralized dataset is an aggregation of all device datasets and have the same split of training and testing data with respect to local settings. The second group is federated learning models which also have two complexity versions (3) FedAvg(ResNet-8), and (4) FedAvg(ResNet-32). We use the most popular FedAvg[7] algorithms to represent this group. The third is (5) FedGKT[8] which is a combination of federated learning and knowledge distillation techniques. All devices use homogeneous ResNet-8 structure and the central server hosts a ResNet-24 structure. The complexity of the coupled network is equivalent to ResNet-32. The last one to compare is (6) D-BCD[9], which utilizes block coordinate decent to perform fully decentralized homogeneous models training. These baselines are iconic and represent the evolution from centralized learning all the way to fully

Table 1 Overall system performance of different methods on Apnea-ECG and Sleep-EEG

	Apnea-ECG				Sleep-EEG			
Methods	Accuracy	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore
Centralized(8)	84.35 ± 0.5	83.58 ± 0.5	81.48 ± 0.6	82.52 ± 0.5	77.64 ± 1.0	72.30 ± 0.8	75.64 ± 0.9	73.58 ± 0.9
Centralized(32)	89.17 ± 0.2	86.93 ± 0.3	85.47 ± 0.3	86.19 ± 0.2	82.13 ± 0.5	74.96 ± 0.4	79.65 ± 0.6	77.06 ± 0.5
FedAvg(8)	78.87 ± 0.7	73.64 ± 0.6	71.47 ± 0.8	72.54 ± 0.6	72.79 ± 1.1	63.62 ± 0.9	66.34 ± 1.3	65.15 ± 1.2
FedAvg(32)	84.79 ± 0.5	82.53 ± 0.6	82.35 ± 0.7	82.44 ± 0.6	77.84 ± 0.8	71.29 ± 0.7	76.45 ± 0.9	73.69 ± 0.8
FedGKT(8+24)	84.74 ± 0.4	82.87 ± 0.3	82.57 ± 0.5	82.72 ± 0.4	78.23 ± 0.9	71.59 ± 0.7	76.66 ± 0.8	74.13 ± 0.7
D-BCD(32)	90.54 ± 0.4	88.23 ± 0.5	85.79 ± 0.3	86.99 ± 0.4	83.44 ± 0.6	76.11 ± 0.6	79.75 ± 0.5	77.89 ± 0.6
DecentKD(Local)	87.92 ± 0.5	85.80 ± 0.5	84.72 ± 0.7	85.26 ± 0.6	81.03 ± 0.9	73.91 ± 0.8	78.66 ± 1.0	76.34 ± 0.9
DecentKD(Layer)	89.23 ± 0.5	87.04 ± 0.6	84.97 ± 0.6	85.99 ± 0.5	82.13 ± 0.8	74.98 ± 0.9	78.99 ± 1.1	76.94 ± 0.8
DecentKD(Unit)	90.97 ± 0.4	90.62 ± 0.4	85.21 ± 0.6	87.83 ± 0.5	83.84 ± 0.6	76.22 ± 0.5	79.31 ± 0.8	77.64 ± 0.7
DecentKD(Unit with PT Loss)	91.19 ± 0.4	88.04 ± 0.4	85.18 ± 0.5	86.69 ± 0.4	84.14 ± 0.7	76.37 ± 0.7	79.28 ± 0.9	77.81 ± 0.8

All methods use standard ResNet1D for processing one dimensional time-series data. Numbers indicate the number of hidden layers in the model. The complexity of DecentKD device models is various and ranges from 8 to 32 sampled by uniform distribution. FedGKT: ResNet-8 on device side and ResNet 24 on server side. D-BCD: block descent based decentralized training algorithm. Local means no collaboration between devices. Layer and unit indicate the decoupling point of feature extractor. PT loss is a loss technique introduced in TinyBERT that calculates deviations between intermediate residual units output. DecentKD based methods can achieve comparable or better performance with centralized or federated methods in a privacy preserved way. The improvement of collaborative DecentKD over silo DecentKD is significant(ρ<0.1). There is significant advantage of unit based decoupling over layer based decoupling(ρ<0.1).

decentralized learning. The metrics used for performance evaluation are Accuracy, Precision, Recall, and Fscore for both tasks. We illustrate the system performance on Apnea-ECG and Sleep-EEG datasets in Table 1.

From the result, we observe that our DecentKD method achieves significant better performance when compared to methods with simple model structure (e.g. Centralized(8), FedAvg(8)) and methods using distillation techniques(FedGKT). Even isolated DecentKD(Local) that does not impose collaboration between devices can outperform models with simple structures. This is because decentralized stored data are mostly Non-IID, global models cannot fit the hard aggregated data well. Decentralized learning is essential for such scenarios to achieve penalization. In addition, our proposed method could also achieve comparable or surpass the performance of methods with deeper networks in terms of accuracy and Fscore. It reveals that DecentKD has the capability to achieve acceptable performance and preserve user privacy simultaneously. For instance, the prediction accuracy of DecentKD(Unit) gets a 1.8-12.1% increase across centralized and partial decentralized methods, and a 0.4% increase to decentralized method for Apnea-ECG detection, as well as a 1.7-11.2% and 0.4% increase for Sleep-EEG diagnosis. Furthermore, the result demonstrates that our dynamic distillation loss control and neighbour selection algorithm based collaboration technique is statistically significantly better than silo decentralized learning, 3.0% and 2.8% accuracy rise, 2.6% and 1.4% Fscore gain are achieved on two tasks. These results verify that our fully decentralized collaborative learning method is effective when privacy protection takes in place.

To better illustrate the advancement of DecentKD, we compare the features it has with other baseline methods in terms of privacy protection and model collaboration in Table 2.

Table 2 Comparison of advanced features DecentKD includes with other methods

Feature	**Centralized**	**FedAVG**	**FedGKT**	**D-BCD**	**DecentKD**
Indirect data share	✗	✓	✓	✓	✓
Model personalization	✗	✗	✓(partial)	✓	✓
No central server requirement	✗	✗	✗	✓	✓
Heterogeneous model support	✗	✗	✗	✗	✓

Existing centralized and federated(not include federated and distillation hybrid method FedGKT) methods cannot achieve model penalization. Decentralized methods(D-BCD, DecentKD) eliminate the need of a central server and utilize on-device learning. DecentKD is more flexible and compatible as it supports heterogeneous model collaboration which best leverages device's capability. DecentKD includes all advanced features to maintain significant performance while protecting user privacy.

Model Effectiveness

Next, we verify the effectiveness of our proposed knowledge distillation based cross device decentralized learning framework.

Discussion

In this work, we propose a decentralized heterogeneous model collaborative learning method named DecentKD, which enables heterogeneous edge devices to deploy models with appropriate size to perform on device learning and inter device collaboration to gain better performance. The main philosophy of DecentKD is exchanging local model's intermediate feature representations with neighbourhood and absorb their knowledge to increase model performance. The method not only tackles the problem of privacy leakage compared to traditional methods that directly share inputs and labels or federated methods that obliquely share weights, but also supports heterogeneous devices to collaborate with each other. Although our approach is mainly verified on e-health datasets, it's not particularly designed for specific tasks. The approach is general and can be applied to many other real world scenarios, especially those have privacy concerns and non-IID data from device to device. It also provides a new direction of how device can collaborate with each other without the need of sharing raw data and can benefit other knowledge transfer based methods in decentralized training environment.

To achieve performance augmentation, we introduce dynamic knowledge distillation loss control based on the soft labels produced by the entire neighbourhood, which helps edge device find optimal neighbour devices to collaborate under non-IID data circumstance. The loss control function also

learns the weighted factor variables through standard local gradient descent which is easy to implemented and can control the amount of knowledge should be learned from neighbourhoods dynamically and effectively.

Different from existing methods such as partial decentralized federated approaches[10][11] and fully decentralized learning[2][9] that only supports homogeneous model collaboration, our DecentKD approach is also compatible with heterogeneous model deployment. Heterogeneous model is more close to the reality. In the real world, devices themselves are naturally of different specifications regarding to their memory size, cpu processing speed, energy consumption, etc. Under normal condition, deeper models that have higher complexity will usually result in better performance, however, for homogeneous on device collaborative learning, one must make sure low rank devices could also fit the model, which could cause overall system performance degrade. DecentKD tackles this problem by only keeping the first simple feature extractor the same. High rank devices can deploy deeper models and make the most of their computing capacity to reach higher performance locally and share the knowledge with others, and low rank devices could run simple models and absorb useful high dimensional information to fine tune its parameters. Each device can find an optimal model structure that best leverage its computing power and memory, as well as learn personalized models[12] to best fit local data distribution. Note that in Table 1, we also include PT loss with the best performed DecentKD method. PT loss[13] is a multi-layer wise loss function that calculates divergence between intermediate layers. Our experiment result shows that it cannot bring significant extra improvement(only 0.22% and 0.20% accuracy respectively on two tasks) compared to simple DecentKD method but impose additional limitation to model structure that reduce overall system heterogeneity.

Many collaborative deep learning frameworks have been developed to enable distributed training[11] on devices' local data. However, most distributed and federated algorithms require the direct share of either raw data or model weights, these types of data transmission will inevitably raise privacy issues. The exposure of personal information may violate certain regulations and cause social problems. The more recent variant of federated learning named FedGKT uses model segmentation to split the whole federated ResNet-64 model into two parts which are local side ResNet-8 and cloud side ResNet-56, they treat the first convolutional layer's output as extracted feature map, and periodically transfer the feature maps between device and cloud to perform federated learning. Based on this work, we introduce feature map based collaborative training in decentralized settings. We also learn from the idea of deep model structure decoupling[14] and find that the best decoupling point for models with branchy stuctures such as ResNet would be unit-wise, which means we would achieve best result if we take entire residual block's output as intermediate feature maps instead of layer output. The result shows 1.7% and 1.6% accuracy gain on Apnea-ECG and Sleep-EEG tasks with first block feature extraction.

In personalized deep learning, not all participating devices in the network could be beneficial to local models. The selection of collaborators is important and can impact overall model effectiveness. In order to solve the personalization problem, we integrate the idea from latent multi-task learning[15], multi-linear relationship quantification[16], and cross-stitch architecture[17] with our algorithm which could find optimal neighbour devices to collaborate and control the amount of knowledge should be transferred at the same time. The proposed method is straightforward and easy to train using the classic gradient descent approach. Our method could improve efficiency especially for large and complex networks while maintain model effectiveness because each device only needs to collaborate with certain number of neighbours.

However, DecentKD is not perfect and has the following limitations. First, DecentKD assumes that each device at least has the amount of local data that could help model converge and achieve certain level of performance. Under some extreme conditions, device model could confront with under fitting problem and fail to learn the parameters. Second, there is a trade-off between communication cost and system heterogeneity. Early block feature extraction could cost more network bandwidth but ensure better system performance while later block feature extraction saves data transmitted but reduce amount of neighbours that can be shared with. Thus, in our future work, we will study how to quantify the trade off and find the best heterogeneity the system should employ. Furthermore, we plan to deploy the framework to other industries that recently imposed regulations on privacy restraints and explore the system effectiveness. Model pruning[18][19] is also a direction that can be integrated to our method, it could help increase computation efficiency and reduce the local model size to support more complex on-device models which gain further prediction accuracy.

Methods

In this section, we introduce the details of our heterogeneous decentralized learning approach based on dynamic knowledge distillation loss control(DecentKD). We first introduce the problem our approach intends to solve, then introduce the way our method works, and finally provide some discussions on penalization.

Problem definition

We denote $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ as the set of all devices in the network, where N is the number of devices. We assume that for a single device $n \in \mathbb{R}^N$ that has M size locally collected dataset $\mathcal{X} = [x_1^n, x_2^n, ..., x_M^n]$ as its input data instances, and $\mathcal{Y} = [y_1^n, y_2^n, ..., y_M^n]$ as its labels. Each device has its unique local model function $f_\theta : \mathcal{X} \to \mathcal{Y}$ that maps local data instances to corresponding labels. Device models are heterogeneous to each other. Local data and labels will never be shared with neighbours. The goal is to learn a strong personalized model for each device collaboratively in a privacy preserved way. The entire training

process does not involve any central server to coordinate the collaboration, communications are established purely through peer-to-peer connections.

Device model

Each device generates a unique local model with the ideal complexity that satisfies its memory constraint. The device model is trainable based on the benchmark of that device's computing power. We achieve model heterogeneity at this stage. Each model uses standard gradient descent method to learn and update its local model parameters for certain number of epochs until there is no further progress in its test loss. We denote the local model loss:

$$\mathcal{L}_n = -\sum_{i=1}^{M}\sum_{c=1}^{P} y^n_{(i,c)} \log(\sigma(h^n_{(i,c)})) \tag{1}$$

where P is the number of classes and $\sigma(h^n_{(i,c)})$ is the softmax activation output of local model's last layer hidden result. Next, we decouple the device model into two major parts: feature extractor Θ_F and local classifier Θ_C. The local model can be written as $f_\theta = argmax(\Theta_C(\Theta_F(\mathcal{X})))$. Note that in our proposed method, all Θ_F would have identical structure which is a standard residual unit to ensure that the intermediate outputs have the same dimension as well as similar information depth. The sparse input data will be encoded to a more dense and abstract representation to maintain user privacy and enable sharing. Θ_C can be heterogeneous from device to device but the input and output dimension will remain the same as all devices' task goal are identical.

Collaboration and dynamic KD loss adaptation

Next, we introduce the details of how DecentKD manages to train ResNet based on-device personalized model collaboratively with the neighbourhood and achieve privacy protection object. In traditional knowledge distillation, devices share input data and soft model output to achieve inter device knowledge transfer. However, direct data share would violate user privacy and cause problem. We choose to share intermediate feature maps which encodes the raw data to high dimensional abstract representations. As the feature extractor is trained privately on each device, one can hardly decode the representation to restore the original data. The local intermediate feature maps are shared periodically through either directly broadcasting to all available devices or publishing to an optional message broker. Neighbour devices calculate soft labels for received anonymous feature maps using local Θ_C and send the result back to the sender. The neighbour device m soft labels with regards to local device n encoded feature maps $h^n_m = \Theta^m_C(\Theta^n_F(\mathcal{X}_n))$. The sender aggregates all valid feedback and perform a stitch operation to calculate the influence of neighbours to its real labels, which is formulated as follows:

$$h'^n = \mathbf{W_n} \cdot [h^n_m, ..., h^n_j] \tag{2}$$

We use this alternative last layer hidden output to calculate the loss using Equation 1 and update the neighbour weight factor matrix $\mathbf{W^n}$. Then we regularize the weight matrix by the following operation:

$$\mathbf{W'_n} = [\frac{|\alpha_m^n|}{\sum_{i=m}^{j}|\alpha_i^n|}, ..., \frac{|\alpha_j^n|}{\sum_{i=m}^{j}|\alpha_i^n|}] \tag{3}$$

where $|\alpha_m^n|$ is the absolute value of the weight of neighbour device m to local device n. Thus the weighted distillation loss is defined as:

$$\mathcal{L}_{KD}^{(m,n)} = \frac{|\alpha_m^n|}{\sum_{i=m}^{j}|\alpha_i^n|} \times KL(\sigma(\frac{h_m^n}{T}), \sigma(\frac{h^n}{T})) \tag{4}$$

where KL is the Kullback–Leibler divergence function. The local main loss is also regularized to:

$$\mathcal{L}'_n = \frac{|\alpha_n^n|}{\sum_{i=m}^{j}|\alpha_i^n|} \times \mathcal{L}_n \tag{5}$$

The final loss is the summation of both the main loss and knowledge distillation losses. Neighbour selection is also achieved by sorting the neighbour weights and keeping the top N. Each device will update its own model parameters using the combined loss through back propagation. The process will be iteratively executed until the model loss does not further decline. The neighbour could choose to complete the training or repeat this process when new neighbours joining into the network. This approach is flexible and fault tolerant, there is no hard requirement of stable connections between devices because the training is asynchronous and each device could re-select or discard neighbours during the training process if losing certain devices.

References

[1] Wu, C., Wu, F., Lyu, L., Qi, T., Huang, Y., Xie, X.: A federated graph neural network framework for privacy-preserving personalization. Nature Communications **13**(1), 1–10 (2022)

[2] Bistritz, I., Mann, A., Bambos, N.: Distributed distillation for on-device learning. Advances in Neural Information Processing Systems **33**, 22593–22604 (2020)

[3] Wu, C., Wu, F., Lyu, L., Huang, Y., Xie, X.: Communication-efficient federated learning via knowledge distillation. Nature communications **13**(1), 1–8 (2022)

[4] Feng, H., You, Z., Chen, M., Zhang, T., Zhu, M., Wu, F., Wu, C., Chen, W.: Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation. In: ICML, pp. 3274–3283 (2021)

[5] Penzel, T., Moody, G.B., Mark, R.G., Goldberger, A.L., Peter, J.H.: The apnea-ecg database. In: Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163), pp. 255–258 (2000). IEEE

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[7] 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). PMLR

[8] He, C., Annavaram, M., Avestimehr, S.: Group knowledge transfer: Federated learning of large cnns at the edge. Advances in Neural Information Processing Systems **33**, 14068–14080 (2020)

[9] Ye, G., Yin, H., Chen, T., Xu, M., Nguyen, Q.V.H., Song, J.: Personalized on-device e-health analytics with decentralized block coordinate descent. IEEE Journal of Biomedical and Health Informatics (2022)

[10] Yao, J., Wang, F., Jia, K., Han, B., Zhou, J., Yang, H.: Device-cloud collaborative learning for recommendation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3865–3874 (2021)

[11] Watcharapichat, P., Morales, V.L., Fernandez, R.C., Pietzuch, P.: Ako: Decentralised deep learning with partial gradient exchange. In: Proceedings of the Seventh ACM Symposium on Cloud Computing, pp. 84–97 (2016)

[12] Schramowski, P., Turan, C., Andersen, N., Rothkopf, C.A., Kersting, K.: Large pre-trained language models contain human-like biases of what is right and wrong to do. Nature Machine Intelligence **4**(3), 258–268 (2022)

[13] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351 (2019)

[14] Li, H., Hu, C., Jiang, J., Wang, Z., Wen, Y., Zhu, W.: Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), pp. 671–678 (2018). IEEE

[15] Ruder, S., Bingel, J., Augenstein, I., Søgaard, A.: Latent multi-task architecture learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4822–4829 (2019)

[16] Long, M., Cao, Z., Wang, J., Yu, P.S.: Learning multiple tasks with multilinear relationship networks. Advances in neural information processing systems **30** (2017)

[17] Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3994–4003 (2016)

[18] Jiang, Y., Wang, S., Valls, V., Ko, B.J., Lee, W.-H., Leung, K.K., Tassiulas, L.: Model pruning enables efficient federated learning on edge devices. IEEE Transactions on Neural Networks and Learning Systems (2022)

[19] Lee, N., Ajanthan, T., Torr, P.H.: Snip: Single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340 (2018)