Differentially Private Knowledge Distillation for Mobile Analytics

Lingjuan Lyu
National University of Singapore
Singapore
lyulj@comp.nus.edu.sg

Chi-Hua Chen
Fuzhou University
China
chihua0826@gmail.com

ABSTRACT

The increasing demand for on-device deep learning necessitates the deployment of deep models on mobile devices. However, directly deploying deep models on mobile devices presents both capacity bottleneck and prohibitive privacy risk. To address these problems, we develop a *Differentially Private Knowledge Distillation* (DPKD) framework to enable on-device deep learning as well as preserve training data privacy. We modify the conventional *Private Aggregation of Teacher Ensembles* (PATE) paradigm by compressing the knowledge acquired by the ensemble of teachers into a student model in a differentially private manner. The student model is then trained on both the labeled, public data and the distilled knowledge by adopting a mixed training algorithm. Extensive experiments on popular image datasets, as well as the real implementation on a mobile device show that DPKD can not only benefit from the distilled knowledge but also provide a strong differential privacy guarantee ($\epsilon = 2$) with only marginal decreases in accuracy.

CCS CONCEPTS

• **Security and privacy** → **Domain-specific security and privacy architectures**; **Privacy protections**; • **Computing methodologies** → **Artificial intelligence**; *Neural networks.*

KEYWORDS

Privacy-preserving; knowledge distillation; deep learning.

ACM Reference Format:
Lingjuan Lyu and Chi-Hua Chen. 2020. Differentially Private Knowledge Distillation for Mobile Analytics. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401259

1 INTRODUCTION

The proliferation of deep learning, especially deep neural networks (DNNs), has led to the notable success in a broad range of applications like image classification, natural language processing, and so on. Meanwhile, service providers are facing a series of efficiency and privacy challenges: (1) running modern DNNs with millions of parameters on computationally limited devices comes with resource limitations and user experience penalties; (2) service providers usually train sophisticated DNN models on the large amount of data collected from users, which commonly contain sensitive information. Directly releasing these models presents significant privacy risks because the adversary can successfully recover sensitive information encoded in these models [4]. Apart from these issues, service providers might be reluctant to share their valuable and highly tuned models due to intellectual restrictions [11].

In order to build small models that can be easily deployed on mobile applications, a number of model compression methods have been proposed, which can be generally categorized into either compressing pretrained networks or training small networks directly [5–7]. Among these methods, knowledge distillation plays an important role [2, 6]. In traditional knowledge distillation, knowledge is distilled from a cumbersome teacher model or the ensemble of teachers to guide the training of a smaller student model with different architecture. It has been verified that this smaller student model can achieve comparable accuracy to the larger teacher model but is less computationally expensive [6]. Despite the encouraging performance of the compressed student model, few model compression methods considered privacy issues [2, 6].

Among the randomization-based techniques [3, 9, 10], only differential privacy (DP) provides theoretic privacy guarantee while incurring small computational cost. To enable knowledge transfer in a differentially private manner, Papernot *et al.* [12, 13] proposed a general framework called *Private Aggregation of Teacher Ensembles* (PATE) for private knowledge transfer, where an ensemble of teacher models were firstly trained on disjoint subsets of the sensitive data, then the aggregator counts votes assigned to each class, adds carefully calibrated Laplacian noise to the resulting vote histogram, and outputs the class with the most noisy votes as the ensemble's prediction. Finally, a student model was trained in a semi-supervised manner using GANs. However, PATE is not aimed for compressing DNNs, a reasonable approach to integrate private knowledge transfer with model compression is yet to be devised.

To overcome the aforementioned challenges, we adapt PATE to output prediction vector to guide the training of a smaller student model more efficiently. Our contributions are summarized as follows: (1) we design a differentially private knowledge distillation framework named DPKD, which embeds the distilled knowledge from an ensemble of teachers into a compressed student model that can be deployed on mobile devices; (2) we propose a mixed training algorithm to ensure high privacy and utility of the student model; (3) we demonstrate the effectiveness of our framework through three popular datasets for real-world mobile analytics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401259

2 PRELIMINARIES

2.1 Model Compression via Distillation

Distillation is designed to extract class probability produced by a large DNN or an ensemble of DNNs to train a smaller DNN with negligible accuracy loss, hence it is useful to train models in service providers before deploying the compressed variants in mobile devices. The general intuition is based on the fact that knowledge acquired during training is not only encoded in model parameters but is also encoded in the probability vectors produced by the DNNs [6]. The benefit of using class probabilities (soft labels) instead of hard labels is intuitive: class probabilities encode additional knowledge about the relative differences between classes compared to a class label. To perform distillation, a large DNN with softmax as output layer is first trained on the original dataset as would usually be done. A softmax output layer converts the logit, z_i, computed for each class into a probability, p_i, by comparing with the other logits. For total c classes, the softmax layer computes each component p_i of the prediction vector $\vec{p}$ as follows:

$$p_i = \frac{exp(z_i/T)}{\sum_{j=1}^{c} exp(z_j/T)} \tag{1}$$

where T is the softmax temperature, which controls the smoothness of probabilities output by the larger model: the higher T, the more ambiguous its probability distribution (*i.e.*, all probabilities of the output are close to $1/c$); similarly, the smaller T, the more discrete its probability distribution (*i.e.*, only one element in the probability vector is close to 1 and the remainder are close to 0). The probability vectors produced by the large DNN are then used to label the training data of the smaller DNN. Afterwards, the smaller DNN is trained at the softmax temperature identical to the one used in the large DNN. The temperature is set back to 1 at test time to produce more discrete probability vectors during classification.

2.2 Differential Privacy

Definition 1. *A randomized algorithm $\mathcal{A}$ satisfies (ϵ, δ)-approximate differential privacy if*

$$\Pr\{\mathcal{A}(D) \in \mathcal{S}\} \leq e^{\epsilon} \Pr\{\mathcal{A}(D') \in \mathcal{S}\} + \delta. \tag{2}$$

for all set $\mathcal{S} \subseteq \text{range}(\mathcal{A})$, and all neighboring datasets D and D', where D and D' differ in one tuple.

Informally, the above definition implies that when ϵ is small enough, the two output distributions are closer to each other, and the algorithm $\mathcal{A}$ becomes more private as it is difficult to determine whether the input is D or D'.

3 PROPOSED FRAMEWORK

3.1 Framework Illustration

Our proposed framework DPKD is presented in Fig. 1, which consists of three main parts:

1) **Teacher ensemble**: An ensemble of n teacher models are trained on partitions of the sensitive, labeled training set. The same softmax temperature T is applied to all teacher models. The sensitive data is only used to train the teacher ensemble which are not accessible to the adversary, insulating the sensitive data from the explicit privacy leakage.

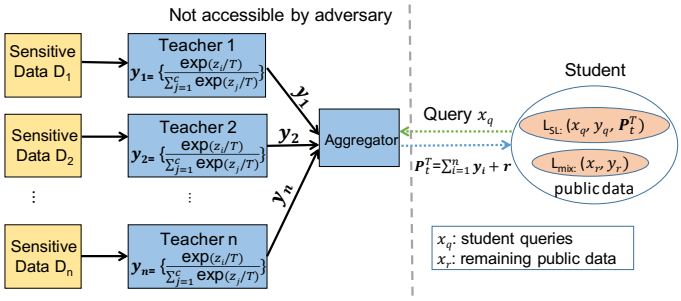


Figure 1: The overview of our DPKD.

2) **Noisy aggregation mechanism**: The distilled knowledge (probability vectors output by teachers) are aggregated and perturbed before normalization. Unlike the original PATE which adds calibrated Laplacian noise to the resulting teacher vote histogram, and outputs the class with the most noisy votes as the ensemble's prediction, we add noise to the aggregated teacher prediction vector, then release the noisy prediction vector. Each element of the aggregated prediction vector can change by at most 1 for a pair of adjacent databases D and D'. To guarantee differential privacy, we follow [13] to add Gaussian noise, which effectively reduces the noise needed to achieve the same privacy cost per student query. In particular, let n and c be the number of teachers and classes, $\vec{y_i}$ be the prediction vector of teacher i, we add random noise vector $\vec{r}$ with each element $r_j \sim \mathsf{N}(0, \sigma^2)$ $(j \in \{1, \cdots, c\})$ to the aggregated prediction vector $\sum_{i=1}^{n} \vec{y_i}$ to ensure that the released noisy aggregation $\sum_{i=1}^{n} \vec{y_i} + \vec{r}$ is differentially private.

3) **Student model**: The final step involves the training of a student model on both the distilled knowledge from teacher ensemble and the labeled public data. To limit the privacy cost of knowledge transfer, only a limited number of queries –denoted by (x_q, y_q) – are answered by the aggregation mechanism.

3.2 Loss Functions for Student Model

Different from the classical use of knowledge distillation, in our framework, we assume the student has access to a portion of public, labeled examples and we focus on supervised learning with knowledge distillation. The knowledge learned from training the ensemble of teachers on the private training set can be distilled and transferred to the student model. This allows the student model to benefit from both the hard class labels and the probability vectors (soft labels) to converge towards an optimal solution. In classic supervised learning, the mismatch between p_s (normal probability generated by the student model with $T = 1$), and y (ground-truth label) is usually penalized using cross-entropy loss as in Equation 3. Hence, for the public samples (x_r, y_r) without knowledge distillation, the student only considers the hard labels by using Equation 3:

$$\mathcal{L}_{SL} = \mathcal{L}_{CE}(p_s, y) \tag{3}$$

In knowledge distillation, the student tries to match the softened outputs of student p_s^T, and teacher p_t^T via a KL-divergence loss under the same temperature T:

$$\mathcal{L}_{KD} = T^2 * \mathcal{L}_{KL}(p_s^T, p_t^T) \tag{4}$$

Therefore, for the queried public samples (x_q, y_q) with knowledge distillation, the student model is trained by using a combination of hard labels and soft labels as per Equation 5:

$$\mathcal{L}_{mix} = \alpha \mathcal{L}_{KD} + (1 - \alpha) * \mathcal{L}_{SL} \tag{5}$$

where the hyperparameter α is used to control the trade-off between the two terms of the loss: the first term forces the optimization towards a similarly softened softmax distribution for the student, while the second term represents the self learning loss, which forces the optimization towards approximating the true label as usual. Intuitively, α being close to 1 means that student puts more "trust" on the knowledge distilled from the teacher ensemble. In particular, $\alpha = 1$ corresponds to only considering the soft labels, while $\alpha = 0$ corresponds to self learning without considering knowledge distillation.

Therefore, the final objective can be written as:

$$\mathcal{L} = \mathcal{L}_{mix} | (x, y) \in (x_q, y_q) + \mathcal{L}_{SL} | (x, y) \in (x_r, y_r) \tag{6}$$

The detailed training process of the student model θ_s is given in Algorithm 1. Since differential privacy is closed under post-processing, the final student model is also differentially private, and safe for publication.

Algorithm 1 Student Model Training

Require: Public data $(x, y) = \{(x_q, y_q), (x_r, y_r)\}$; training epochs T; batch size B; iterations in each epoch $R = |x|/B$.
  for $r \in \{1 : T\}$ **do**
    for $t \in \{1 : R\}$ **do**
      Sample a batch x_B of size B from x;
      Compute mixed loss $\mathcal{L} = \mathcal{L}_{mix} | x_B \in x_q + \mathcal{L}_{SL} | x_B \in x_r$;
      Backpropagate $\mathcal{L}$ to update θ_s;
    end for
  end for

4 EXPERIMENTS

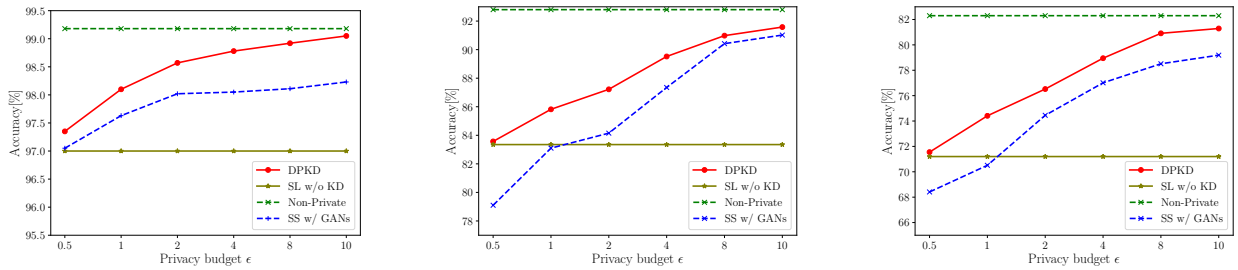
Experiment setup. Our experiments are conducted on three popular image datasets: MNIST[1], SVHN[2], and CIFAR-10[3]. For each dataset, all the training set are used as private data to train teacher ensembles. The test set is split into two halves: the first is used as labeled inputs to simulate the student's public data and the second is used as a hold out to evaluate test performance. For MNIST, we randomly choose 9,000 samples from the test set as the public data to train the student, among which a subset of samples is selected as queries. We evaluate the performance of the student model on the remaining 1,000 samples of the test set. Similarly, we randomly choose 10,000 and 9,000 samples from the test set to train the student models for SVHN and CIFAR-10, and test on the remaining 16,032 and 1,000 samples respectively. Moreover, for query sample selection, we follow [13] to use Confident-Gaussian NoisyMax (GNMax) aggregator, which allows teachers to filter out queries for which teachers do not have an overwhelming consensus, thus avoiding answering expensive queries and allowing more queries to be answered within a fixed privacy budget. We use the ensembles of 250 teachers for all datasets, and follow [13] to adopt the state-of-the-art privacy accountant *Rényi Differential Privacy* (RDP) to track the privacy loss spent on answering the queries made to the teacher ensemble until the pre-allocated privacy budget is used up, this allows the training of the follow-up student model with meaningful privacy bounds. We remark that RDP can support the exact numerical computations for a finite number of orders, avoiding the need to specify a discrete list of moments ahead of time as required in the moments accountant approach used in [12], delivering tighter privacy bounds.

For model architecture, three widely used convolutional deep neural networks, *i.e.*, Conv-Small, Conv-Middle, and Conv-Large [8, 14, 15], are used for MNIST, SVHN and CIFAR-10 respectively. We use Conv-Small as the student model, and compress it to a smaller size for the study of compression performance. During student model learning, we use grid search of certain empirical values of α and T in Equation 5 to choose the suitable hyperparameter combinations for different datasets. The training epochs and batch size are set as 100 and 64 for both mixed learning and self learning.

Performance evaluation. We first compare the performance of our DPKD with three baselines: (1) a self-learned student model trained without knowledge distillation; (2) a non-private teacher model learned with the entire training set; (3) a semi-supervised student model with GANs [12]. As indicated in Figure 2, for MNIST, when the student model is self-learned on all the labeled public data without knowledge distillation, the student model has a 97% test accuracy. In contrast, the accuracies of the student models using DPKD are consistently higher than that of the self-learned student models, this observation also applies to the other two datasets. We also observe that these accuracies generally rise with the privacy budget ϵ, this is because higher privacy budgets allow more queries to be answered, *i.e.*, more information from the teacher ensemble is transferred to the student. Moreover, even when the distilled knowledge is protected by a strong privacy budget $\epsilon <= 2$, the student can still largely benefit from the knowledge distillation. In addition, we simulate a more strict scenario, where the public data does not contain any samples of a highly sensitive class. For example, the self-learned student model can never recognise the samples from class 1 if its pubic data does not contain any samples of class 1. On the other hand, DPKD can largely improve the student's accuracy on the class 1 at the cost of paying for a reasonable privacy loss even though the student never encountered class 1 during training. All the above results validate that the knowledge distillation from teacher ensemble indeed helps the student model generalize better.

We also note that when all the public samples are unlabelled, the student model can only train on a limited number of queries labeled by the ensemble in a supervised manner. In this case, the accuracies of the student models on three datasets would be much lower, even when the student models are trained using semi-supervised learning with GANs [12, 13]. This is because the teacher ensemble cannot produce high quantity and quality labels for the unlabelled public samples, especially at low ϵ values. This leads to the poor performance of the follow-up supervised or semi-supervised training of the student model. All these observations manifest that student training on the labeled public data in combination with the knowledge distillation from the teacher ensemble can help the student model avoid overfitting to the limited data [1].

[1] http://yann.lecun.com/exdb/mnist/
[2] http://ufldl.stanford.edu/housenumbers/
[3] https://www.cs.toronto.edu/~kriz/cifar.html



(a) MNIST (b) SVHN (c) CIFAR-10

Figure 2: Accuracy vs. privacy budget ϵ over three datasets. SL w/o KD: self-learned student model without knowledge distillation; Non-Private: non-private teacher model learned with the entire training set; SS w/ GANs: semi-supervised learning of student model with GANs (PATE).

We next validate the compression performance in Table 1. For MNIST, SVHN, and CIFAR-10, we set the privacy bound of (ϵ, δ) as $(2, 10^{-5})$, $(2, 10^{-6})$, and $(2, 10^{-5})$, respectively. In order to examine the runtime on mobile devices, we follow [15] to experiment on HUAWEI HONOR 8 with ARM Cortex-A72@2.3GHz and Cortex-A53@1.81GHz to process 100 images. Since the whole parameters of the teacher ensemble are at least n times larger than the student model, which is impractical to deploy on mobile devices, so we use a non-private teacher model for comparison.

On all the three datasets, even though the student models are of much less capacity and fewer training data, training with DPKD can help the student models obtain comparable accuracies to the non-private teacher models. For example, on MNIST, the student achieves 23× compression ratio and 19× speed-up with only 0.36% accuracy decrease. Moreover, we state that the student model can achieve better performance at the cost of using larger models. All these results validate that DPKD can be effectively used to privately compress large models with acceptable accuracy loss.

Table 1: Compression performance on mobile phones (NPT=Non-private Teacher, S=Student).

Dataset	Model	Performance		
		#Params	Time[s]	Acc[%]
MNIST	NPT	155.21K	0.76	99.48
	S	6.54K	0.04	99.12
SVHN	NPT	5.86M	7.38	96.45
	S	0.09M	0.59	95.02
CIFAR-10	NPT	12.89M	12.20	86.48
	S	1.25M	4.65	85.15

5 CONCLUSION AND FUTURE WORK

In this paper, we develop a differentially private knowledge distillation (DPKD) framework to enable on-device deep learning as well as preserving training data privacy. The knowledge acquired by the ensemble of teachers, acting as the large model, is compressed into a smaller student model in a differentially private manner. A novel training algorithm is proposed to train the student model with high privacy and utility. Experiments through three image datasets on mobile devices indicate that the student model can yield a comparable accuracy even under a rigorous privacy guarantee. Future work will focus on further refinement of the student model to improve both accuracy and privacy. It is expected that our framework can be generalized to a wide spectrum of applications, as our framework is model agnostic to the underlying machine-learning techniques.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (Nos. 61906043, 61877010, 11501114 and 11901100), Fujian Natural Science Funds (No. 2019J01243), Funds of Fujian Provincial Department of Education (No. JAT190026), and Fuzhou University (Nos. 510730/XRC-18075, 510809/GXRC-19037, 510649/XRC-18049 and 510650/XRC-18050).

REFERENCES

[1] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Advances in neural information processing systems*. 2654–2662.
[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of KDD*. ACM, 535–541.
[3] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
[4] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM CCS*. ACM, 1322–1333.
[5] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).
[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
[7] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
[8] Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *ICLR*.
[9] Lingjuan Lyu, James C Bezdek, Yee Wei Law, Xuanli He, and Marimuthu Palaniswami. 2018. Privacy-preserving collaborative fuzzy clustering. *Data & Knowledge Engineering* (2018).
[10] Lingjuan Lyu, Yee Wei Law, Sarah M Erfani, Christopher Leckie, and Marimuthu Palaniswami. 2016. An improved scheme for privacy-preserving collaborative anomaly detection. In *PerCom Workshops*. IEEE, 1–6.
[11] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, Jiong Jin, Han Yu, and Kee Siong Ng. 2020. Towards Fair and Privacy-Preserving Federated Deep Models. *IEEE TPDS* 31, 11 (2020), 2524–2541.
[12] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2017. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*.
[13] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *ICLR*.
[14] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. 2018. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of AAAI*.
[15] Ji Wang, Weidong Bao, Lichao Sun, Xiaomin Zhu, Bokai Cao, and S Yu Philip. 2019. Private model compression via knowledge distillation. In *Proceedings of AAAI*. 1190–1197.