DOI: 10.1561/9781638284772.ch7

Chapter 7

Private Deep Learning

By Nicolas Papernot

Because they learn features from data directly, rather than rely on human engineering, deep neural networks and the associated *deep learning* algorithms continue to see increased adoption from machine learning practitioners. This creates new information flows involving data analyzed in systems deploying deep learning in particular when this data is used to train the neural networks. This data is often about individuals and can be sensitive in nature in application domains like healthcare or language modeling. To prevent models from leaking private information included in their training data, it is thus important to develop deep learning algorithms that are respectful of their training data's privacy.

In this chapter, we introduce two classes of approaches to train deep neural networks with differential privacy, an established framework to reason about the privacy of algorithms. The first is a variant of a popular learning algorithm—stochastic gradient descent. The second does not require any modifications of the learning algorithm and instead obtains privacy through postprocessing of the model's outputs. We conclude with a discussion of practical aspects of deploying deep learning with differential privacy as well as current research problems.

7.1 Introduction

The intuition behind deep learning is to learn hierarchical representations of data [LBH15]. Deep learning algorithms rely on deep neural networks to model

data, where a deep neural network is obtained by stacking multiple layers of neurons—each layer representing one abstraction of the data. Classical approaches to machine learning typically rely heavily on the features chosen to represent the data at the input of the model in order to extract a mapping between the input features and the output. For instance, the scale-invariant feature transform [Low99] was used in several areas of computer vision to represent images at the input of a model while encoding priors such as the fact that feature extraction should be invariant to translation or rotation of the image. Instead, deep learning promises to alleviate any human feature engineering and instead extract useful representations directly from the data, in conjunction to learning the mapping between inputs and outputs [LBH15; GBCB16]. These representations are learned by a deep neural network's different layers. Intuitively, one expects representations to be increasingly more abstract as one goes through the model's hierarchy of layers: eventually, we'd like the representation to be sufficiently abstract that it disentangles the different factors of variation so that we can predict the desired output. For instance, in a classification problem we would like the last representation output by a model to linearly separate the different classes of the problem. Deep learning makes representation learning easier by sequentially composing representations: the model starts by learning a first representation from its inputs, this first representation is then itself used as an input to learn a second representation, and so on, until we obtain the model's last representation which is used to infer the model output.

Deep learning enabled many promising developments in the machine learning community. The first successes of deep learning saw applications to computer vision, and in particular object recognition [KSH12]. Since then, these techniques have been applied to other vision problems including in healthcare to diagnose certain conditions from xray images of human chests [Irv+19; CHBB20]. Another domain is the one of language modeling, with the ability to translate between natural languages being a prominent application [BCB14].

Because of the sensitive nature of data analyzed in these applications, it is important to reason about the privacy of deep learning algorithms. Take the example of a machine learning model trained to predict whether a patient was readmitted to a hospital shortly after being dismissed. Being part of the training set for such a model is itself private information: individuals may wish to keep secret the fact that they were hospitalized. In this chapter, we present research which demonstrates that an adversary can use the model's prediction to infer whether a particular patient record's was included to train the model (see Section 7.2.1 on membership inference). This jeopardizes the privacy of individuals who contributed their medical records to the model's training set and could decrease their willingness to consent the sharing of their medical record in the future. In another example, imagine that a machine learning model is trained to model the English language on a corpus of

private correspondence. Such a model could for instance be used to predict the next words being typed by a user on a smartphone keyboard. If a user types "My address is", we would like to ensure that the model does not complete the sentence with the address of a different user. These two examples surface the need for an approach to deep learning which provides strong guarantees of privacy for the training data. Here, we focus on the framework of differential privacy to reason about such guarantees because it is the most established framework for this purpose. Differential privacy allows us to quantify the degree of privacy protection provided by an algorithm on the underlying (sensitive) data set it operates on. The reader is referred to Chapter 1 for a review of the notion of differential privacy and its properties. That said, differential privacy is not a silver bullet for all privacy-related questions. We discuss these other aspects of privacy that fall outside the scope of differential privacy in Section 7.6

Overview of the Chapter

In this chapter, we introduce two approaches for deep learning with differential privacy: differentially private stochastic gradient descent (see Section 7.4) and the private aggregation of teacher ensembles (see Section 7.5). Before we do so, we first cover attacks against the privacy of training data used to train deep neural networks (Section 7.2). For clarity of exposition, we focus on deep learning approaches for classification in a supervised setting. In other words, the outputs of our models are always chosen among a discrete set of *classes* (e.g., a set of objects for object classification in computer vision).

7.2 Privacy Attacks against Deep Learning

It is often wrongly assumed that a model that generalizes well preserves privacy. One of the reasons this is not true is that generalization guarantees are average-case whereas privacy needs to be provided for everyone, thus it is a worst-case guarantee. The best way to illustrate this is with attacks that show how deep learning algorithms can leak private information contained in their training set, especially so when the dataset contains outlier that deviate from distributional assumptions made about the data.

The canonical privacy attack against deep learning is membership inference (see Chapter 5). In this attack, the goal of the adversary is simply to infer whether a given data point was included in the model's training set or not [SSSS17]. This may be seen as a weak attack against the private data used to train the model—given that it requires that the adversary already know the data point and its attributes prior to the attack—but recall that certain applications of deep learning include healthcare.

Imagine a model is trained on data obtained from patients who all had a certain medical condition, e.g., cancer, to predict their reaction to certain drugs based on morphological information. In this case, being part of the dataset is itself a sensitive attribute which an individual may not wish to reveal.

Furthermore, the membership inference adversary is also very much in line with the definition of differential privacy, which as we will see in greater detail later in this chapter, asks that the output of a differentially private learning algorithm not significantly change when a single training record is added, modified, or deleted [DMNS06]. This makes membership inference an interesting adversary to have in mind if one wishes to evaluate the strength of the privacy guarantees provided by an algorithm that claims to be privacy-preserving. Indeed, stronger guarantees of differential privacy (as evidenced by analytically proving a smaller upper bound on the privacy leakage ε) imply a lower success rate of membership inference. We will discuss work in this vein later in Section 7.7.2, after we have introduced algorithms for deep learning with differential privacy in Sections 7.4 and 7.5.

More sophisticated attacks will seek to recover missing attributes of a data record already partially known to the adversary. For instance, an adversary may already know the name, age, and zip code for an individual but would like to recover another attribute, e.g., the individual's blood type, included in the record analyzed by the machine learning model. Ultimately, an adversary would like to extract entire training points from the deep learning model. We will detail how such an attack was mounted on a language model.

7.2.1 Membership Inference

Membership inference attacks were covered in details in Chapter 5, we review them here within the notation considered in this chapter for completeness.

In membership inference, the adversary is given the capability of querying the model for predictions on any input of their choice. Shokri et al. introduce the first approach for membership inference against deep learning [SSSS17]. Let us write x the input which the adversary would like to know whether it was in the training set or not to train a model h. The authors propose that the adversary train a model to solve the membership inference task. The adversary will train this second machine learning model m to take in as its input the predictions $h(x)$ of the first model, that is the victim model, on the input x of interest to the adversary and output a binary label $m(h(x)) \in \{0, 1\}$ indicating whether or not x was used to train h. Note that membership inference attacks were initially introduced with the assumption that the adversary can not only observe the victim model's predictions but also its confidence on these predictions. Put another way, the victim prediction $h(x)$ is a

confidence vector indicating the likelihood that the input x belongs to each of the classes of the task.

Of course, the adversary is unable to access the data used to train h. As a consequence, they cannot directly collect a set of prediction vectors on inputs in and out of the victim model's training set to train the binary membership inference classifier m from this set. Instead, the adversary will first seek to produce a collection of shadow models h_i whose training set is known to the adversary. In Shokri et al., the authors train shadow models from the same distribution than the one used to sample training data for the victim model.[i] However, it was later shown that this data only needs be from the same domain as the victim model's training data [Sal+18]. These shadow models h_i can then be used to train the membership inference classifier m because the adversary has access to their training set and can thus collect a dataset of prediction vectors $h_i(x)$ for inputs x which are both *in* and *out* of the dataset used to train h_i.

Why is training the membership inference classifier m possible? Intuitively, this is because the confidence scores output by the shadow models—and later the victim model—vary from training to non-training points. In fact, this difference can be so important that Shalem et al. [Sal+18] show that simply thresholding the confidence score for the label predicted by the victim model is sufficient for the adversary to have better than random chance at inferring membership: the adversary simply predicts that a point was in the training set when the prediction is associated with a high confidence score, and conversely points with low confidence predictions are predicted to be outside the training set. In the limit, the adversary could further simplify the attack to predict that a point is a member of the training set if and only if it is correctly classified [YGFJ18]. Put another way, the membership inference classifier m of Shokri et al. can be though of as calibrating this confidence threshold automatically by observing a collection of predictions made on points in and out of the training set. Training a membership inference classifier also has the added benefit that this classifier can learn from any privacy leakage induced by the confidence scores associated with the other classes of the problem which do not end up being predicted—as opposed to the approach of Shalem et al., which focuses on the label receiving the largest score.

More recently, Choquette-Choo et al. [CTCP20] innovate over the initial work of Shokri et al. to propose an attack strategy that does *not* require that the adversary have access to the confidence scores output by a model. To obtain additional private leakage to what was obtained previously by simply guessing that correctly-classified inputs were members of the training data (see supra

i. This is simulated in their experiments by splitting the training set in multiple disjoint sets.

on [YGFJ18]), Choquette-Choo et al. make *multiple* queries to the victim model to infer membership inference of a *single* input. These queries are crafted to approximate the confidence of the victim model; the adversary compares the perturbation magnitude a point can sustain before it is classified by the victim model with a different label. The larger that perturbation, the more confident the victim model is in predicting on this input. In turn, the more likely this input is to be a member of the training set. Perturbations introduced to measure this proxy of confidence can vary from domain to domain. The authors first consider data augmentations typically used to regularize training in the machine learning literature, such as translations and rotations in the image domain. When little is known about the domain, the adversary may proceed by inserting isotropic Gaussian noise to inputs but also perturbations computed through gradient descent to approximate as closely as possible the distance to the victim model's decision boundary—as would be done to find an adversarial example.[ii] Intuitively, this attack exploits the fact that training points are further away from the victim model's decision boundaries. While attacks described above exploited confidence scores to measure this distance in the output domain of the victim model, Choquette-Choo et al. thus measured this distance directly in the input domain by comparing the victim model's predicted class on multiple inputs between the point of interest (for membership inference) and the model's decision boundary. Thus, the resulting strategy is a *label-only* membership inference attack, but it comes at the expense of (possibly significantly) higher query complexity.

We refer the reader to Chapter 5 for a more in-depth discussion of membership inference attacks, including refinements over these initial attack methodologies.

7.2.2 Attribute Inference and Training Data Extraction

Membership inference is not equally successful on all training points. Certain points may be more susceptible to membership inference because they are outlier to the training distribution; Yeom et al. analyze the connection between membership inference and overfitting [YGFJ18]. They report that overfitting is sufficient but not necessary for membership inference, as we will discuss in more details later in Section 7.2.3. This difference in the worst-case and average-case perspectives can lead to further privacy leakage where signal from the model can be exploited not

ii. Adversarial examples are inputs crafted by an adversary to force a model to misclassify. Most algorithms craft adversarial examples by performing a form of gradient descent from an originally correctly classified input towards one of the model's decision boundaries surrounding it [Big+13; Sze+13]. When instead the adversary operates without access to the model, in a black-box setting, this requires that the adversary either obtain a copy of the victim model [Pap+17], playing a similar role to the shadow models of Shokri et al. in the context of membership inference, or that the adversary make multiple queries to perform zero-order optimization [Che+17] and recover gradients through finite differences.

only to perform membership inference, but also to recover information about the training data which was initially unknown to the adversary: e.g., missing attributes of a training point or even the training point itself altogether. Next, we describe one approach for each of these two adversarial goals. Both approaches have in common that they reduce the problem or recovering information unknown to the adversary to the problem of membership inference.

Attribute Inference

Whereas membership inference assumes the adversary already has access to the data point they are interested in testing the membership of, attribute inference instead assumes the adversary only has partial access to this data point. The adversary desires to recover missing attributed from this data point.

For instance, Yeom et al. mention the problem of an adversary with partial access to a medical record who wishes to recover some of the missing attributes of this patient's record [YGFJ18]. The authors introduce an approach which reduces the problem of attribute inference to membership inference. The adversary randomly guesses a value for the missing attribute and performs membership inference on the resulting data point. If the point is accepted as a member of the training set, the adversary assumes that their random guess for the value of the missing attribute was correct. Otherwise, the adversary makes a new guess and repeats the membership inference test. Yeom et al. note that the adversary will be further advantaged should they have any knowledge of the distribution of attribute values which would help them improve their guessing strategy. This observation is also exploited by the work we describe next below.

Extracting Training Data

Indeed, Carlini et al. [Car+20] further extend Yeom et al.'s attack strategy to perform training data extraction from a large language model known as GPT-2 [Rad+19]. The attack strategy is also an extension of the membership inference attacks described previously; it further assumes that the adversary has access to a generative model, that is a model which can be used to sample data points from the underlying data distribution. Carlini et al. use the victim model itself since it is a generative language model. The attack strategy is as follows: the adversary first randomly generates a large pool of data points from the input domain of the victim model, then these inputs are ranked by decreasing likelihood with or without the help of the language model, finally membership inference is done on inputs ranked highest with the victim model this time. Similar to Yeom et al., the crux of the attack is thus for the adversary to sample points—using the language model—that are more likely to be training points, which then reduces training data extraction

to the problem of membership inference. Generating such points may require significant adversarial knowledge. Here, this is evidenced by the adversary's access to the victim model itself and to knowledge about its training data collection process. Nevertheless, the authors show how this simple attack strategy is able to extract personal information about one individual from GPT-2's training set, which is a large corpus of data crawled from the Internet.

7.2.3 Take-aways from the Attacks

Research on attacks is motivated by the need to inform defenses. The different attacks we presented above all point to an intricate relationship between overfitting and privacy leakage.

Membership Inference and Overfitting

Recall that membership inference is not equally successful on all training points. Does that mean that overfitting is necessary to obtain a membership inference signal? Although the initial work of Shokri et al suggested that this was the case [SSSS17], later the analysis of Yeom et al. shows that overfitting is sufficient but not necessary [YGFJ18]. Yeom et al. obtain this result by formalizing the gain an adversary obtains from membership inference on stable classifiers which provably do not overfit. We note however that this adversary remains largely an analytical tool and that instantiating it in practice may not always be possible because it is assumed that the adversary colludes with the learning algorithm to obtain a model on which membership inference is successful despite the lack of overfitting. It is also possible for overfitting to facilitate white-box membership inference, that is attacks that inspect a model's parameters. For instance, Leino and Fredrikson [LF20] leverage the idiosyncratic use of features by a model to improve membership inference success. Taking a step back, these results imply that points who are outlier to the training data distribution but are still included in the training set are more susceptible to membership inference—because the model will have overfitted to memorize the correct prediction for these outliers [Lon+18]—but that no point is safe from the privacy leakage associated with membership inference.

The Need for Differential Privacy

Perhaps the single most important take-away from these attacks on the privacy of training data in deep learning, and in particular from results on membership inference, is that simple defense mechanisms such as obfuscating the model's confidence [SSSS17] or regularizing training through an ℓ_2 objective penalty [SSSS17; NSH18; Jia+19] or dropout [Jia+19] are not sufficient to prevent leakage of private information. Furthermore, attacks vary greatly in the assumptions they make about the adversary's knowledge and capabilities, suggesting that it is difficult for

the defender to foresee the (auxiliary) knowledge which an adversary will exploit to leak private information from the model [NS08]. Results we have presented thus far hence point to the need for a worst-case framework to reason about the privacy of training algorithm. This is precisely what differential privacy promises. Thus, we next turn to two approaches which can train deep neural networks while providing differential privacy guarantees that bound how much private information leaks from the training data.

7.3 How to Obtain Differential Privacy in Deep Learning?

Training deep neural networks with differential privacy is challenging because the loss function minimized during learning is non-convex. This rules out approaches like objective perturbation [CMS11; KST12] introduced in the convex setting. The two approaches to deep learning with differential privacy we present in this Chapter follow the same design principle.[iii] First, the learning procedure is approximated by a sequential composition of computations whose sensitivity is known. Second, noise is added to these computations. Finally, the total privacy loss of the resulting learning procedure is analyzed to determine the differential privacy guarantees it provides. While we present the two approaches in the context of training deep neural networks, they can also be used for convex learners and can sometimes outperform approaches tailored to the convex setting [Iye+19].

As discussed in Section 1.4.1 of Chapter 1, an essential step to design a differentially private algorithm is to estimate its sensitivity to neighboring input datasets. This is a prerequisite to calibrate noise which is added to the algorithm's computations. Measuring sensitivity of learning algorithms used to train deep neural networks is not an easy task given that the model's architecture yields a non-convex optimization problem to set the model's parameter values.

In the following two Sections, we describe two approaches which differ in the way that they control the sensitivity of the learning algorithm. The first approach named DP-SGD (see Section 7.4) proposes to modify the model updates computed by stochastic gradient descent, in order to achieve a differentially private stochastic gradient descent. At a high level, modifications are two-fold: the gradients are first clipped to control sensitivity, and then they are noised to obtain differential privacy. The second approach named PATE (see Section 7.5) instead proposes to have an ensemble of models trained without privacy predict with differential

iii. Note that the first approach injects noise within the algorithm's computations while the second approach injects noise to the algorithm's outputs. Differential privacy can also be obtained by injecting noise to the algorithm's inputs, but this class of approaches is out of scope here because it is not specific to deep learning.

privacy by having these models predict in aggregate rather than revealing their individual predictions. Because each model only contributes part of the final prediction, the sensitivity of the inference procedure can be controlled and differential privacy obtained.

In the remainder of this Section, we establish terminology and concepts which will be instrumental in understanding the privacy analysis of both DP-SGD in Section 7.4 and PATE in Section 7.5. These concepts allow us to reason about the privacy loss of each computation performed by the learning algorithm, as well as accounting required throughout learning to derive privacy guarantees post hoc.

7.3.1 Reasoning about the Privacy Loss

When developing the analysis of randomized algorithms to prove that they are differentially private, it can be useful to observe that it is equivalent to show that a randomized algorithm is differentially private and that a tail bound can be established on the randomized algorithm's privacy loss. Here, the privacy loss of a randomized algorithm $\mathcal{M}$ is a random variable defined as:

Definition 7.1 (Privacy Loss)**.** *Let $d, d' \in \mathcal{D}^n$ be neighboring datasets. For an auxiliary input aux and outcome $o \in \mathcal{R}$, we define the privacy loss of the randomized mechanism $\mathcal{M}$ at o to be:*

$$c(o, \mathcal{M}, aux, d, d') = \log \frac{Pr[\mathcal{M}(aux, d) = o]}{Pr[\mathcal{M}(aux, d') = o]}$$

In the context of deep learning, where the randomized algorithm is the training procedure, this allows us to reason about the privacy of one access to the training data. To capture multiple accesses to the training data (e.g., each step of a stochastic gradient descent), it is necessary to compose the privacy loss incurred at each data access. Coming back to the attacks we described in Section 7.2, this is intuitive because a model trained for longer is given more opportunities to memorize private information contained in its training set. Unfortunately, directly composing tail bounds on the privacy loss can significantly loosen the overall bound thus resulting in meaningless guarantees of differential privacy for learning. Thus, several modern approaches including Abadi et al. [Aba+16] propose to instead analyze the moments of the privacy loss random variable.

7.3.2 The Moments Accountant

Key to the privacy analysis of both DP-SGD and PATE (which we will present in Sections 7.4 and 7.5 respectively) is the moments accountant, a stronger accounting

method to accumulate the privacy loss expended by the learning algorithm each time it accesses the training data. Different from tail bounds on the privacy loss, bounds on the log moments of the privacy loss compose *linearly*. Reasoning over moments of the privacy loss random variable enables the moments accountant to yield a much tighter analysis of the privacy guarantees of learning algorithms which are composed of a sequence of differentially private mechanism applications.

Abadi et al. define the λ^{th} moment as the log of moment generating function evaluated at the value λ.

Definition 7.2 (λ^{th} moment)**.** *The λ^{th} moment for the privacy loss of the randomized mechanism $\mathcal{M}$ is:*

$$\alpha_{\mathcal{M}}(\lambda, aux, d, d') = \log \mathbb{E}_{o\sim\mathcal{M}(aux,d)}[\exp(\lambda \cdot c(o, \mathcal{M}, aux, d, d'))].$$

Because a data-independent differential privacy guarantee should hold for all possible pairs of neighboring datasets, the analyses of DP-SGD and PATE look at bounding all possible λ^{th} moments for a given mechanism $\mathcal{M}$, regardless of the auxiliary input *aux* or the pair of datasets d, d'. This yields the following variable α of interest:

$$\alpha_{\mathcal{M}}(\lambda) = \max_{aux,d,d'} \alpha_{\mathcal{M}}(\lambda, aux, d, d'). \tag{7.1}$$

Abadi et al. prove two properties of α which are instrumental in the moments accountant ability to offer a tighter privacy analysis of DP-SGD and PATE. First, they show that the log moments of the privacy loss random variable composes linearly. This is useful to analyze randomized algorithms like DP-SGD and PATE which—as we will explain later—can be viewed as a sequential application of k differentially private mechanisms $\mathcal{M}_i$. In the differential privacy literature, such a sequential application is referred to as adaptive composition: the auxiliary input *aux* of the k^{th} mechanism $\mathcal{M}_k$ is the output of all the previous mechanisms.

Theorem 7.3 (Composability of α)**.** *] If a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_k$ where $\mathcal{M}_i : \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \to \mathcal{R}_i$, then for any λ:*

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{i=1}^{k} \alpha_{\mathcal{M}_i}(\lambda). \tag{7.2}$$

We will later see how in the case of both DP-SGD and PATE, this composability property is used to bound the moment of the total privacy loss incurred by the learning algorithm across all of its accesses to the training data. Once this accounting is complete, a value of ε for the differential privacy is derived from the bound on α through an application of the following tail bound.

Theorem 7.4 (Tail bound on α)**.** *] For any ε, the mechanism $\mathcal{M}$ is (ε, δ)-differentially private for:*

$$\delta = \min_{\lambda} exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon). \tag{7.3}$$

7.4 Differentially Private Stochastic Gradient Descent (DP-SGD)

The work of Abadi et al. [Aba+16] on DP-SGD builds on a line of work [SCS13; BST14], which was itself initiated by the pioneering work of Chaudhuri et al. [CMS11]. One of its key innovations is the introduction of the novel analysis technique known as the moments accountant (see Section 7.3) to improve the privacy guarantee which one can prove after training a deep neural network with DP-SGD. We first introduce DP-SGD in its most modern form, and then outline a sketch for the analysis of its privacy guarantees in Section 7.4.2.

7.4.1 The DP-SGD Algorithm

Let us first describe a typical *non-private* stochastic gradient descent. To begin, one samples a minibatches of training data, that is randomly draw a small number[iv] of training examples (X, Y) from the training set where X are the inputs and Y the labels associated with these examples. One then computes the loss $l(\theta, X, Y)$ of the model h, averaged over all examples contained in the minibatch. The loss measures the error between the model's predictions and the labels we expected it to produce. A common choice for multi-class classification problems is the cross-entropy, which computes for a single training example (x_i, y_i):

$$l(\theta, x_i, y_i) = -\sum_j y_{ij} \log h_j(x_i),$$

where j indexes the classes of the classification problem, and i indexes the different examples included in the minibatch such that $(X, Y) = \{(x_i, y_i)\}_i$. Next, one computes the gradient of the average loss across the minibatch with respect to the model's parameters. Such a gradient can be computed because deep neural networks

iv. This number of training examples included in a minibatch, i.e., its *size*, is chosen first and foremost to accommodate the hardware computing gradient descent. For instance, if the machine is equipped with an accelerator for matrix algebra, e.g., a GPU, it is generally-speaking beneficial to use the largest minibatch size which will fit in the accelerator's memory. However, other factors may be relevant in the choice of minimatch size and may commend a smaller size; smaller minibatches were for example observed to introduce a regularization effect in learning although this effect is not consistently obtained [Sha+18]. Thus, in practice one treats the minibatch size as a hyperparameter of stochastic gradient descent.

are differentiable, and this computation is made more efficient by the backpropagation algorithm [RHW86]. This gradient is multiplied by a constant α called the learning rate before it is subtracted to the current model parameter θ to obtain a step of gradient descent:

$$\theta \leftarrow \theta - \alpha \frac{\partial \frac{1}{N} \sum_i l(\theta, x_i, y_i)}{\partial \theta}, \tag{7.4}$$

where N is the size of the minibatch. The learning rate is a hyperparameter controlling the size of each step of gradient descent. To bootstrap this procedure, one initializes the model parameters θ to a random value. The steps described above are then applies repeatedly until the deep neural network's parameters are deemed to have converged. Ideally, one would like the model to converge to the global minimum of the loss function it is trained to minimize. However, deep neural networks being non-convex learners, one must be satisfied with one of the local minima that approximates the global minimal loss value. In practice, the stopping criterion is typically decided by evaluating the performance of the model on a holdout set of data known as the validation set. This validation accuracy should generally increase as steps of gradient descent are taken. Once this validation accuracy plateaus or starts to decrease (indicating overfitting [Zha+16]), gradient descent is interrupted.

The modifications made by Abadi et al. [Aba+16] to obtain a differentially private stochastic gradient descent algorithm are three fold. First, all operations within the minibatch are performed on a *per-example* basis. This means that the loss is no longer averaged across examples included in the minibatch. Instead, the loss $l(\theta, x_i, y_i)$ of each training example (x_i, y_i) is computed individually. The same holds for gradients of the loss with respect to the model parameters:

$$g_i = \frac{\partial l(\theta, x_i, y_i)}{\partial \theta}.$$

The second key modification made is to clip each of these per example gradients to have a maximum ℓ_2 norm C:

$$g_i \leftarrow g_i \cdot \min\left(1, \frac{C}{\|g_i\|_2}\right). \tag{7.5}$$

Intuitively, this bounds the influence of each training example on learning, and more important this bound is known to be C. We can thus derive the sensitivity of the learning algorithm from the value C of this bound. This is essential to the third, and last, modification introduced to obtain differential privacy: Gaussian noise whose standard deviation is calibrated to be linearly proportional to C

is added to each of these clipped gradients.[v] The rest of the procedure is unmodified: the average of noisy clipped gradients is multiplied by the learning rate and subtracted to the model parameters to update the model. Abadi et al. thus obtain the following gradient descent step:

$$\theta \leftarrow \theta - \alpha \frac{1}{N} \sum_i \left(\frac{\partial l(\theta, x_i, y_i)}{\partial \theta} \cdot \min\left(1, \frac{C}{\|g_i\|_2}\right) + N(0, \sigma^2 C^2) \right), \quad (7.6)$$

where σ is a hyperparameter of the differentially private stochastic gradient descent. Generally speaking, the larger the value of σ is, the tighter the resulting differential privacy guarantee is (i.e., one can prove a smaller bound on the privacy loss ε).

7.4.2 Privacy Analysis of DP-SGD

The analysis of DP-SGD relies on composition theorems to accumulate the privacy budget spent by gradient descent across multiple steps; each step is accounted for and further increases the privacy cost of learning. In addition to the application of advanced composition theorems to compute the overall privacy budget, one can also refine the analysis of DP-SGD by reasoning about its guarantees with definitions like zero-concentrated differential privacy [BS16] or Renyi differential privacy [Mir17]. For simplicity, we expose here the analysis provided by Abadi et al. [Aba+16]. It derives the differential privacy guarantees obtained by training with DP-SGD from an application of the moments accountant.

We begin by analyzing the λ^{th} moment for the privacy loss of an *individual* step of gradient descent. Recall that each step of DP-SGD noises the clipped gradient computed over a minibatch J of examples chosen independently at random with probability q. Thus a step of DP-SGD can be written as $\mathcal{M}(d) = \sum_{i \in J} f(x_i, y_i) + N(0, \sigma^2)$ where $f(x_i, y_i)$ computes the clipped gradient for example (x_i, y_i) and σ here incorporates the clipping norm C for simplicity of presentation. The authors show that under certain conditions (see the paper), the following upper bound holds:

$$\alpha_{\mathcal{M}}(\lambda) \leq \frac{q^2 \lambda(\lambda + 1)}{(1 - q)\sigma^2} + O(q^3 \lambda^3 / \sigma^3).$$

v. Note that adding noise to gradients is commonly done in deep learning to regularize learning [Nee+15]. Here, it is however crucial to first clip gradients before they are noised, and then to calibrate the standard deviation of noise added to the clipping bound. Otherwise, one would be unable to prove that the resulting procedure is differentially private.

They then leverage the composability of a sequence of adaptive mechanisms (the individual steps of gradient descent) to obtain the λ^{th} moment $\alpha(\lambda)$ for the privacy loss of the *complete* gradient descent. If we denote by T the total number of steps of gradient descent completed, the following is derived by an application of Equation 7.2:

$$\alpha(\lambda) \leq \frac{Tq^2\lambda^2}{\sigma^2}.$$

Through an application of the tail bound on the moments for the privacy loss (see Equation 7.3) gives the main result of Abadi et al., which is that DP-SGD is (ε, δ)-differentially private for any $\varepsilon < c_1 q^2 T$ and $\delta > 0$ if we choose:

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\varepsilon}, \tag{7.7}$$

for some constants c_1 and c_2.

In practice, it is interesting to note that interpreting this analysis and the guarantees it provides can be non-trivial. First, the analysis assumes that each minibatch is obtained by *independently* uniformly sampling training examples. Instead, many practical implementation of stochastic gradient descent only analyze *permutations* of the training set at each epoch so they must be carefully extended to obtain an valid implementation of DP-SGD. Second, the analysis leverages randomness in the sampling of each minibatch to guarantee differential privacy. This result is known as privacy amplification by subsampling in the literature [BBG18]. However, this requires that the randomness is of cryptographic strength, which again is not always the case in all practical implementations. We come back to additional practical considerations for implementing DP-SGD in Section 7.6.

7.5 Private Aggregation of Teacher Ensembles (PATE)

DP-SGD obtains differential privacy by modifying the learning procedure itself to bound its sensitivity and randomize it accordingly. This means that any deviations from the learning procedure captured by the existing theoretical analysis requires new efforts to bound the privacy leakage. This may limit the ability of DP-SGD to new developments in the machine learning community (although it has proven itself remarkably amenable to a wide range of learning tasks beyond classification and variants of SGD like the widely-used Adam [KB14] optimizer). A good example is the one of batch normalization, now a prevalent technique in modern deep

learning [IS15], thanks to its ability to regularize the norm of internal model activations which can help prevent overfitting and improve convergence of the optimization procedure. However, batch normalization involves the computation of statistics across all examples included in a minibatch. This means that it cannot be used in conjunction with DP-SGD because privacy guarantees rely on the gradient computation being performed on a per-example basis. This motivates the need for another privacy-preserving approach to deep learning which does not limit how the model is trained.

Papernot et al. [Pap+16; Pap+17] introduce such an approach with the private aggregation of teacher ensembles (PATE). Rather than modify the learning algorithm used to train a model, PATE introduces changes to how data is fed to the learning procedure and how model predictions are revealed.

7.5.1 The PATE Approach

In PATE, one starts by splitting the training set whose privacy needs to be protected in k partitions. These are partitions in the mathematical sense, so one point from the original training set is included in exactly one of the partitions (which are non-overlapping and do not repeat any of the points). From each of these partitions, one trains a machine learning model. There are no restrictions on how each of the models are trained. This means we can for instance use techniques such as batch normalization, whose privacy would be difficult to analyze if we wanted to take an approach in the vein of DP-SGD. In PATE, each model could even be trained with different techniques (e.g., deep learning, decision tree, etc.). Because one trains a model on each of the k partitions, we obtain an ensemble of k models trained independently but all solving the same task. We give the name of *teacher* to each of these k models. Next, PATE deploys this ensemble of teachers to predict on test inputs while bounding leakage of private information from the training data.

PATE obtains privacy by having these teachers predict collectively and revealing their aggregate prediction rather than their individual predictions. Specifically, each teacher is given the input[vi] x that we would like to predict on and commits a vote for one of the labels of the problem. One then builds a histogram of these votes, where each bar $n_j(x)$ of the histogram corresponds to a label j of the problem and indicates how many teachers voted for this label j. The label predicted by PATE is the one which received the most number of votes, i.e., the argmax of this histogram:

$$\arg\max_j n_j(x).$$

vi. To avoid ambiguity, let us stress that all teachers simultaneously predict on the same input here.

Such a mechanism is common in machine learning, even when one does not seek to provide guarantee, because bagging predictors is an instance of ensemble learning which commonly improves generalization [Bre96].

Intuitively, it can be seen that when most teachers agree on the label to be predicted, revealing the most frequent label does not leak much private information in the sense that this prediction was made independently by many teachers. Given that each of these teachers learned from a different set of data, a single training record contained in the original training set (prior to partitioning) could have only influenced one of the teachers, and as a consequence had a negligible impact on the computation of the most frequent label. That said, this voting mechanism is not sufficient to obtain differential privacy because there still exists certain edge cases where a single training point can change the outcome of voting. To see why, observe that if we have an ensemble of k teachers such that k is an odd integer and that $n_a(x) = \frac{k-1}{2} + 1$ teachers voted for label a while the remaining $n_b(x) = \frac{k-1}{2}$ teachers voted for another label b when they are presented with a test input, then a single teacher changing its prediction from label a to b can result in the voting mechanism outputting b rather than a as would have been the case initially.

This shows that in the worst case one teacher alone is able to change the outcome of the voting mechanism. Because each teacher is trained without any form of privacy, we thus have to assume that changing one of the training examples (in the original set before it is partitioned) can change the predictions of the teacher trained on the partition containing this example. Put altogether, this implies that the sensitivity of the mechanism is such that each training record can in the worst case affect at most 2 of histogram bars by $+/-1$ votes. This allows us to calibrate the noise that we need to add to the histogram before we obtain a differentially private prediction. In fact, privately releasing the argmax of a histogram is a well-known problem in the differential privacy literature and the properties of such *histogram queries* are well understood [DR+14]. Specifically, the resulting voting mechanism for PATE computes the following:

$$\arg\max_j \left\{ n_j(x) + Lap\left(\frac{2}{\gamma}\right) \right\}, \tag{7.8}$$

where $Lap(b)$ is the Laplacian distribution with location 0 and scale b.

7.5.2 Privacy Analysis of PATE

A first privacy analysis of the variant of PATE we presented in Section 7.5.1 can be derived from the differential privacy of the Laplace mechanism for histogram queries. Indeed, consider a single prediction made by PATE. This prediction is a histogram query, which is known to be (γ, δ)-differentially private [DR+14]. Next, we

turn to the analysis of a sequence of T predictions made by PATE. Given that each of these predictions is differentially private, we can compose their privacy guarantee to obtain the privacy guarantee achieved overall when making the T predictions. If we naively compose, this results in the T predictions being $(T\gamma, T\delta)$-differentially private. This however yields poor utility when computing Equation 7.8 for most practical values of γ and T. Thus, we turn to the moments accountant we introduced in Section 7.3.2.

Applying the Moments Accountant

Recall that the moments accountant instead proposes to reason in the log space of the moment generating function of the privacy loss. For PATE with the Laplace mechanism, we have that the privacy loss is equal to the noise parameter of the Laplace distribution itself [DR+14]: $c(o, \mathcal{M}, aux, d, d') = \gamma$. From Equation 7.1, we then have that:

$$\alpha_{\mathcal{M}}(\lambda, aux, d, d') = \log \mathbb{E}_{o\sim\mathcal{M}(aux,d)}[\exp(\lambda\gamma)] = \lambda\gamma.$$

We compute these for several integer values of $\lambda \geq 1$. We can then use the tail bound from Definition 7.4 to derive the (ε, δ)-differential privacy guarantee achieved:

$$\varepsilon = \frac{\min_\lambda \alpha_{\mathcal{M}}(\lambda) - \log\delta}{\lambda}. \tag{7.9}$$

Still, this may result in poor utility for practical values of γ and T. There are several ways one can improve the utility-privacy trade-off of PATE. Of course, one can obtain better bounds on the log moments of the privacy loss. For instance, Bun and Steinke [BS16] allow us to show that we also have $\alpha_{\mathcal{M}}(\lambda) \leq \frac{1}{2}\gamma^2\lambda(\lambda+1)$. We can thus retain the best of the two bounds for each of our log moment computations:

$$\alpha_{\mathcal{M}}(\lambda) = \min\{2\lambda\gamma, \frac{1}{2}\gamma^2\lambda(\lambda+1)\}.$$

Data-dependent Analysis

Up to now, our analysis of PATE has been data-independent; that is, our privacy analysis does not depend on the particular dataset which the learning algorithm takes as an input. This is made clear by the maximum operator over aux, d, d' when computing $\alpha_{\mathcal{M}}$ in Equation 7.1. However, in many cases the teachers in the ensemble exhibit a strong consensus on the label they predict: that is one of the labels is assigned most of the votes in the histogram computing by the voting mechanism in PATE. When this is the case, the privacy cost is much smaller than what is suggested by the data-independent analysis.

To obtain such a data-dependent privacy analysis of PATE, Papernot et al. [Pap+16] prove a bound of the log moment of the privacy loss which depends on the probability q that the noisy argmax mechanism will output the wrong label after noising the votes originally cast by teachers:

$$\alpha_{\mathcal{M}}^{dd} = \log\left((1-q)\left(\frac{1-q}{1-e^{\gamma}q}\right)^{\lambda} + qe^{\gamma\lambda}\right). \tag{7.10}$$

The probability q can be derived from the votes cast by teachers as follows:

$$q \leq \sum_{j \neq j^*} \frac{2 + \frac{\gamma}{2}(n_{j^*} - n_j)}{4exp(\frac{\gamma}{2}(n_{j^*} - n_j))}. \tag{7.11}$$

The rest of the privacy analysis is identical to its data-independent counterpart.

In practice, the data-dependent analysis may not always be superior to the data-independent analysis. This is however not a drawback given that we can combine the two effectively at the level of the log moment computation. Indeed, one can compute $\alpha_{\mathcal{M}}$ as follows:

$$\alpha_{\mathcal{M}}(\lambda) = \min\{2\lambda\gamma, \frac{1}{2}\gamma^2\lambda(\lambda+1), \alpha_{\mathcal{M}}^{dd}\}. \tag{7.12}$$

To summarize, the combined data-independent and data-dependent analysis is as follows:

1. Compute the probability q that the noisy argmax will output a prediction that is not the most frequent vote cast by teachers.
2. Compute the three bounds (two data-independent, and one data-dependent based on the value of q) on the log moments of the privacy loss.
3. Retain the minimum bound as in Equation 7.12 for each moment.
4. Derive the (ε, δ)-differential privacy guarantee from the tail bound property of the log moment, as in Equation 7.9.

With data-dependent analysis, we are able to prove a tighter bound on the privacy loss and thus the same mechanism (with constant utility) became "more" private. This however should be interpreted carefully because the guarantee now depends on the dataset which the algorithm is operating on (rather than holding for any possible dataset) and releasing the value of ε now leaks private information. Thus, it is possible to noise the value of data-dependent ε estimates to prevent any additional private information leakage (see Papernot et al. [Pap+16] for a detailed discussion and an example approach for doing so based on the smooth analysis of Nissim et al. [NRS07]).

Training a Student Model

Even with the data-dependent analysis, the privacy budget increases every time a query is answered by PATE. That is, the privacy leakage induced by responding to each query is bounded but non-zero. Thus, the mechanism will either be able to answer a small number of queries only or provide little utility (because the noise injected will have to be larger to decrease the per-query cost of a larger number of queries to be answered). To alleviate this limitation, the privacy-preserving labels predicted by the teacher ensemble can be use to train a student model. This student model can query teachers with unlabeled *public* data. The student will then learn from the labels predicted by the teachers. Once the student is trained, the total privacy budget expended is fixed and the teachers can be discarded. The student can then answer as many queries as it wants, this will not impact the bound. To further improve the utility-privacy tradeoff, the student can also be trained with a semi-supervised learning approach [Sal+16; Ber+19].

7.6 Practical Considerations for Private Deep Learning

Having introduced two approaches to deep learning with differential privacy, DP-SGD and PATE, we now outline a few practical considerations and observations stemming from experience implementing both of these approaches.

7.6.1 Tuning Hyperparameters and the Utility-Privacy Tradeoff

The first obstacle is the accuracy of privacy-preserving models. Datasets are often sampled from a distribution with heavy tails (see Chapter 10). For instance, in a medical application, there are typically (and fortunately) fewer patients with a given medical condition than patients without that condition. This means that there are fewer training examples for patients with each medical condition to learn from. Because differential privacy prevents us from learning patterns which are not found generally across the training data, it limits our ability to learn from these patients for which we have very few examples of [SPGG21]. More generally, there is often a trade-off between the accuracy of a model and the strength of the differential privacy guarantee it was trained with: the smaller the privacy budget is, the larger the impact on accuracy typically is. That said, this tension is not always inevitable and there are instances where privacy and accuracy are synergistic because differential privacy implies generalization (but not vice versa) [Dwo+15]. When there is a tradeoff, it can be partially mitigated by carefully setting the hyperparameters that control both approaches.

As far as DP-SGD is concerned, clipping and noising are common regularizer in machine learning so the fact that DP-SGD may be potentially harmful to

performance can be counter-intuitive. That said, there are key differences in how clipping and noising are used to obtain DP-SGD. First, clipping is typically done on the *average* gradient computed over an entire minibatch. Instead, DP-SGD applies clipping on a *per-example* basis. Work has began understanding the effect this has on learning [STT20], but it remains an open problem to mitigate its impact on the accuracy [Pap+20] of training. Similarly, the impact of noising on DP-SGD remains to be understood. Another important hyperparameter in DP-SGD is the size of the minibatch. Practical experience shows that larger minibatches often yield better tradeoffs between utility and privacy [De+22]. That said, possible values for the size of the minibatch are often limited by hardware constraining the computational feasibility of calculating gradients for a large number of training examples simultaneously. Finally, given that the privacy guarantee of DP-SGD is proportional to the number T of gradient descent steps (recall Equation 7.7), it is important to tune this number and adjust accordingly the learning rate to compensate for a smaller number of steps being taken.

In the case of PATE, the key hyperparameter to be tuned is the number of teachers k to use. Increasing the number of teachers generally improves the privacy guarantee. Because more votes are being aggregated, it is easier to introduce more noise in the aggregation without affecting the outcome of the aggregation (i.e., degrading utility). However, increasing the number of teachers generally has diminishing returns beyond a certain point because each teacher receive so little data to learn from that it becomes a *weak learner*. Thus, there is often in practice an optimal number of teachers to be picked by training teachers with a partition of data whose size varies.

7.6.2 System Perspective on Privacy

Abadi et al. [Aba+17] remark that the privacy of training data may depend on other stages of the data life cycle. Here, we have not discussed techniques for sanitizing the data. These often take the form of removing data attributes which may identify an individual, a class of technique known as data anonymization. We chose not to discuss these approaches because their limitations are well-understood in the privacy community. Fundamentally, anonymization assumes the data owner can anticipate which information the adversary will have access to so that they can deduce with attributes may identify an individual. In other words, anonymization is highly context-dependent and does not provide nearly as strong guarantees as those expressed in the framework of differential privacy. Anonymization may however be beneficial at the data collection stage in certain settings, in particular to minimize the amount of data that is retained about an individual. Indeed, this may be beneficial in settings where proper access control is not enforced when accessing the

training data or its byproducts like machine learning models. We discuss this further in Section 7.6.4. Finally, Abadi et al. note the importance of mechanisms for data deletion and data retention policies. This has been put forward in several legislative frameworks, most prominently in the European Union's General Data Protection Regulation [Man13]. Deleting training data raises the question of what should be done to machine learning models trained on this data. Indeed, even if these models were trained with differential privacy, they would still be influenced by the training data which was deleted (although if the model was trained with differential privacy, this influence would be bounded). Consequently, research has began towards enabling machine unlearning, that is the process of deleting any influence a training point may have had on the machine learning model's parameters. Research on machine unlearning initially focused on learning algorithms which can be expressed in the statistical query learning framework [CY15]. However, this approach does not apply to deep learning because deep neural networks are trained using adaptive statistical query algorithms, which make queries that depend on all queries previously made. Instead, work by Bourtoule et al. [Bou+21] proposes an approach for machine unlearning that is applicable to any model trained with stochastic gradient descent—thus including deep learning.

7.6.3 Public Data

It is interesting to note that the utility of both DP-SGD and PATE can be improved with the availability of public data. We already covered in Section 7.5 how public data is required to train a student model in PATE. Without such public data, the privacy budget of PATE would be unbounded over time (i.e., it would increase each time the teachers answer an additional query). This would of course lead to a disadvantageous tradeoff between utility and privacy. Instead, with public data it is easier to improve this tradeoff because the student can learn with very little supervision from the teachers. The less supervision required, the stronger the privacy guarantee achieved. Here, it is interesting to note that any progress made in the machine learning community towards learning with less supervision can be directly applied to improve the utility-privacy tradeoff in PATE, as demonstrated with the invention of MixMatch [Ber+19].

Models trained with DP-SGD can also be improved with public data: for instance, models can be pre-trained or fine-tuned with public data. Pre-training is now a common practice in machine learning with privacy [LTLH21; De+22]. Fine-tuning is an alternative to pre-training. Intuitively, fine-tuning has the advantage that the private model was already trained so one can exploit this information to select the public data which will most help alleviate the deficiencies of the private model [Zha+19].

We will come back to the issue of data complexity in Section 7.7.1, where we will see that the benefits of public data are a symptom for a more fundamental question in private deep learning around the (im)possibility of learning features from limited data with privacy.

7.6.4 Confidentiality vs. Privacy

In computer security, confidentiality is achieved when information is only accessible to parties who are authorized to access it [Bis02]. In the context of data involved in deep learning algorithms, this means that the training and test data should only be visible to authorized users. This is a guarantee which is orthogonal to the one of differential privacy, where we instead want to prevent inferences about the data (even inferences that can be made without direct access to the data).

To provide confidentiality, one must apply cryptographic primitives so as to encrypt the data and require a key to decrypt the corresponding plaintext. To enable applications of deep learning to encrypted data, some proposed homomorphic machine learning techniques [Gil+16]. These are however difficult to scale to deep neural networks because of the non-linear activation functions they include [TB18]. These techniques can be leveraged to train the model without decrypting data (i.e., perform gradient descent over encrypted data). Cryptography can also be used to make predictions over encrypted test inputs. This is useful when one wants to offload compute associated with the model (e.g. to a cloud provider) without revealing the test inputs (e.g., patient records collected by a hospital).

It is important to note that confidentiality does not imply privacy, or vice versa. This confusion arises frequently, in particular in the context of distributed learning. For instance, federated learning [Kon+16] is an approach for distributed learning with confidentiality but it does not provide any privacy guarantee (in the sense of differential privacy) unless it is combined with an optimizer like DP-SGD. If DP-SGD is not employed, federated learning may leak private information more so than a centralized learning algorithm because it exposes the intermediate model updates [MSDS19; Boe+23b; Boe+23a]. To obtain both confidentiality and privacy, one can also combine cryptographic primitives with the PATE approach from Section 7.5. This yields a form of collaborative learning where the different teachers can be distributed [Cho+21].

7.7 Research Issues

In the rest of this chapter, we present some of the questions currently being investigated by the research community. This is done to give the reader a sense of the

limitations of current approaches, at a more abstract level than the practical considerations of Section 7.6. The list of questions covered here is by no means meant to be comprehensive.

7.7.1 Is Representation Learning possible with Privacy?

Representation learning is fundamental to deep learning. In essence, the intuition behind training neural networks with multiple hidden layers is precisely to have each of these layers extract a representation of the data. The last layer of a deep neural network can be thought as a linear model taking in as its input a representation whose features were learned by composing all of the previous layers of the deep neural network. This alleviates the need for human feature engineering, as demonstrated by the successful application of deep learning to raw images or audio. That said, the trade-off empirically observed between model utility and privacy raises the question of whether current approaches to deep learning can indeed learn representations while satisfying the constraints of differential privacy.

A Feature Perspective

Put another way, is the deep learning promise of not having to human engineer features compatible with the constrains of privacy. Indeed, Tramèr and Boneh [TB20] remark that private deep learning often falls short, in terms of *prediction accuracy*, to shallow models trained without privacy. Building on this observation, they compare the performance of private deep learning with private *shallow* learning combined with human-engineered features. They for instance find that it is possible to simultaneously outperform the accuracy and privacy of private deep learning on CIFAR10 by training a linear model whose features are human engineered. In their experiment, they extract these features with a scaterring network [OM15] which is a non-learned feature extractor. They find that one of the roadblocks preventing DP-SGD from achieving representation learning that outperforms shallow learning from human-engineered features is that the privacy guarantees require that each training point be only involved in a small number of updates. This is a roadblock to deep learning of representations with differential privacy, because deep learning often involves long training procedures taking multiple passes through the dataset (i.e., epochs) to uncover the features which project data on a representation that mostly[vii] linearly separates the data. Tramèr and Boneh then outline the increased data complexity of private deep learning, surfacing the need for larger datasets to

vii. Not all datasets will be linearly separable, and here we are referring to the split in deep neural network architectures described above: the last layer can be interpreted as a linear model taking in as input the features learned by the rest of the architecture.

enable representation learning with privacy. The reader may recall our related discussion in Section 7.6 on public data. This observation is also consistent with a prior successful application of DP-SGD to language modeling in a setting where the data collected spans billions of users [MRTZ17]. Alternatively, Tramèr and Boneh propose to tackle this increased data complexity through transfer learning, as previously studied in the non-private setting [KSL19].

An Architecture Perspective

Having discussed the role that features play in private deep learning, it is natural to consider next the role played by the model's architecture. Deep neural networks are often the fruit of an extensive empirical architectural design phase, combined with the necessary hyperparameter tuning. This is done empirically and involves a large number of runs. It is thus tempting to reuse existing architectures known to perform well and train them, with privacy this time, on datasets that are sensitive. This avoids the cost of hyperparameter tuning with privacy guarantees, which would involve composing the cost of the training run associated with each hyperparameter value that was considered [PS21]. However, Papernot et al. [Pap+20] show how reusing hyperparameter values that were optimal for the non-private setting to now train with a private optimizer can lead to suboptimal architectural choices because the deep neural network is designed for optimal convergence with a non-private optimizer like SGD rather than with a private optimizer such as DP-SGD.

7.7.2 Is the Analysis of DP-SGD (or PATE) Tight?

In the following, we focus on DP-SGD for clarity of exposition, but the discussion is also relevant to PATE. Because the framework of differential privacy involves a rather strict definition of what it means for an algorithm to be private, and in particular that this definition is *worst-case*, it is legitimate to ask whether the guarantees obtained by training with DP-SGD are sufficiently strong, even if the bound on the privacy cost ε is too large to be meaningful in terms to probabilities. Indeed, in theory any value of ε above 1 is not meaningful because the value of ε is used to create an interval between two probabilities (which naturally take values between 0 and 1). This is an important issue in practice because practitioners are faced with the fact that they need to calibrate the standard deviation of Gaussian noise added to gradient descent in DP-SGD with the desired strength of differential privacy guarantee. Recall that increasing the linear factor σ generally leads to smaller values of ε. While it is easy to say that a model is more private than another model by comparing the values of ε we are able to prove (assuming both models are trained using an algorithm whose privacy analysis is tight), it is more difficult to provide an absolute threshold after which we consider an algorithm to be "private". This is of

course an important issue which merits attention from legislative bodies interested in creating new privacy regulations based on the framework of differential privacy.

There are two ways in which one could improve the trade-off between utility and differential privacy in the context of deep learning. First, one could realize that the privacy analysis of DP-SGD is loose, leading to an overestimate of the worst-case privacy leakage. This would mean that efforts towards improving our theoretical understanding of DP-SGD could lead to tighter analysis of its privacy guarantees, and as a by-product to improved trade-offs between utility and differential privacy. Such refined analysis could also require that one make additional assumptions restraining the adversary's capabilities. Second, it is possible that we could invent novel algorithms for learning with differential privacy that afford between utility-privacy trade-offs. These algorithms could be improved variants of DP-SGD or completely new approaches for learning with differential privacy.

It is thus tempting to evaluate the performance of differentially private models in the face of adversaries mounting attacks such as the ones presented in Section 7.2. This would help assess how "far" the analytical guarantees one can prove are from the empirical guarantees of privacy. One should of course be careful about making such an evaluation because the conclusion one makes are very different from the conclusions derived from the analysis of differential privacy guarantees. Put simply, differential privacy guarantees provide an upper bound on the privacy leakage, whereas mounting an attack provides a lower bound on this privacy leakage. Indeed, the adversary is only able to demonstrate that they can extract at least as much private information as they could under that specific attack and threat model. Nothing prevents a more sophisticated adversary from extracting more private information if they are able to gain additional knowledge in the future or discover a new attack strategy. Instead, the guarantees provided by a specific analysis technique for differential privacy hold regardless of the knowledge and capabilities available to the adversary as long as these comply with the definition of differential privacy. Furthermore, this also means that in practice while the analytical bound obtained is fixed given a specific analysis technique, the results of an empirical evaluation of the success of a privacy attack or of the model utility may vary greatly. As a consequence, one should take great care when comparing the privacy-utility tradeoff obtained through analytical guarantees and the privacy-utility tradeoff measured through an empirical evaluation of attack success [EMRS19].

With all of these caveats stated clearly, let us now turn our attention to studies that have recently evaluated the robustness of models trained with differential privacy in the face of attacks targeting the privacy of training data. While these attacks cannot replace the analytical bounds discussed above, they can nevertheless inform theoretical research on these bounds. Jagielski et al. [JUO20] take a first step towards such an evaluation by instantiating a poisoning adversary to audit the

guarantees provided by DP-SGD. They observe that if a model is ε-differentially private, an adversary should have bounded success of at most $\frac{e^{k\varepsilon}}{1+e^{\varepsilon}}$ in telling whether a model was trained on a dataset D or a second dataset $D \cup S$ where S is a set of k poisoning points. To help the adversary succeed as closely as possible as tolerated by the differential privacy guarantee, Jagielski et al. propose a novel attack which minimizes the variance of model parameters when crafting poisoning points.

Building on these results, Nasr et al. [Nas+21] introduce a spectrum of adversaries ranging from the black-box adversary of Jagielski et al. to a more sophisticated (and perhaps unrealistic) adversaries. This enables Nasr et al. to obtain stronger lower bounds on the privacy leakage. This also has the added benefit of singling out the adversarial capabilities needed to match the maximal privacy leakage tolerated by the analytical upper bound on privacy.

7.7.3 Connection Between Privacy and Robustness

Because differential privacy is a worst-case guarantee that implies a form of generalization, it is natural to ask whether learning with differential privacy also nurtures a form of robustness. Machine learning algorithms are indeed known to be vulnerable to both training and test time manipulation of data, the former being referred to as a poisoning [BNL12] attack while the latter is known as an adversarial example [Big+13; Sze+13]. We should be careful when drawing connections between privacy and robustness because results from the differential privacy literature show improvement in average-case generalization whereas robustness requires worst-case guarantees: an adversary will always choose the attack that is the most effective.

We have already outlined the connection between differentially private learning and poisoning in Section 7.7.2. This connection has led Ma et al. to study how differential privacy can be used to defend against poisoning attacks against ridge and logistic regressions [MZH19].

Now take the case of adversarial examples. They are crafted by exploiting the excessive sensitivity of machine learning models to input-domain perturbations. Such perturbations can be found by gradient descent, using the same techniques and tools than those required to train the model (e.g., backpropagation for gradient descent). Song et al. have first noted that improving robustness of models to adversarial examples may lead to additional leakage of private information [SSM19]. This can be understood intuitively with the example of adversarial training, a common technique to improve a model's robustness to adversarial examples: it encourages the model to be a more constant predictor around its training inputs. This means in turn that a membership inference attack is easier to mount because training points are characterized by smoother regions of the loss surface. Research has also investigated whether techniques from the adversarial example literature can

be leveraged to contain privacy leakage from a model [NSH18; Jia+19], but these generally provide empirical guarantees which do not have the properties that make differential privacy attractive (e.g., it is agnostic to adversarial knowledge or capabilities).

7.8 Concluding Remarks

In this Chapter, we first motivated the need for privacy-preserving approaches to deep learning. In addition to needs for privacy inherent to fostering user trust and complying with regulation in place, attacks leaking private information from the training data of machine learning models demonstrate that learning algorithms retain information specific to individual training points. We then presented two approaches for privacy-preserving deep learning, DP-SGD and PATE. While DP-SGD is currently the de facto approach for private deep learning, PATE has potential to shine when it comes to training large model architectures or when predicting in distributed settings from a collection of models. In the rest of the Chapter, we presented challenges that practitioners will likely face but also more fundamental questions that researchers are currently tackling to improve our understanding of how to learn with privacy. While deep learning has enabled key innovations in the past years, deep learning with privacy is still in its infancy. Addressing its current limitations will be key to enable responsible applications of artificial intelligence in areas like healthcare where the potential benefits to society are immense but the risks are commensurately important. Achieving privacy-preserving deep learning by design, rather than attempting to retrofit existing algorithms with privacy considerations is perhaps one of the most promising avenues for future work in this area.

Acknowledgments

The author would like to thank Abhradeep Guha Thakurta for helpful discussions and co-authoring a blog post on a similar topic which helped preparing the exposition of ideas presented here.

References

[Aba+16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. “Deep Learning with Differential Privacy”. In: Proceedings of the 2016 ACM SIGSAC Conference on

Computer and Communications Security. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 308–318. ISBN: 9781450341394. URL: https://doi.org/10.1145/2976749.2978318 (cit. on pp. 260, 262–264).

[Aba+17] M. Abadi, U. Erlingsson, I. Goodfellow, H. B. McMahan, I. Mironov, N. Papernot, K. Talwar, and L. Zhang. "On the protection of private information in machine learning systems: Two recent approches". In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). IEEE. 2017, pp. 1–6 (cit. on p. 271).

[BBG18] B. Balle, G. Barthe, and M. Gaboardi. "Privacy amplification by subsampling: Tight analyses via couplings and divergences". In: arXiv preprint arXiv:1807.01647 (2018) (cit. on p. 265).

[BCB14] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate". In: arXiv preprint arXiv:1409.0473 (2014) (cit. on p. 252).

[Ber+19] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. "Mixmatch: A holistic approach to semi-supervised learning". In: arXiv preprint arXiv:1905.02249 (2019) (cit. on pp. 270, 272).

[Big+13] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. "Evasion attacks against machine learning at test time". In: Joint European conference on machine learning and knowledge discovery in databases. Springer. 2013, pp. 387–402 (cit. on pp. 256, 277).

[Bis02] M. A. Bishop. The art and science of computer security. 2002 (cit. on p. 273).

[BNL12] B. Biggio, B. Nelson, and P. Laskov. "Poisoning attacks against support vector machines". In: arXiv preprint arXiv:1206.6389 (2012) (cit. on p. 277).

[Boe+23a] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot. "Reconstructing Individual Data Points in Federated Learning Hardened with Differential Privacy and Secure Aggregation". In: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P). IEEE. 2023, pp. 241–257 (cit. on p. 273).

[Boe+23b] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot. "When the curious abandon honesty: Federated learning is not private". In: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P). IEEE. 2023, pp. 175–199 (cit. on p. 273).

[Bou+21] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. "Machine unlearning". In: Proceedings of the 42nd IEEE Symposium on Security and Privacy, San Francisco, CA. 2021 (cit. on p. 272).

[Bre96] L. Breiman. "Bagging predictors". In: Machine learning 24.2 (1996), pp. 123–140 (cit. on p. 267).

[BS16] M. Bun and T. Steinke. "Concentrated differential privacy: Simplifications, extensions, and lower bounds". In: Theory of Cryptography Conference. Springer. 2016, pp. 635–658. URL: https://arxiv.org/abs/1605.02065 (cit. on pp. 264, 268).

[BST14] R. Bassily, A. Smith, and A. Thakurta. "Private empirical risk minimization: Efficient algorithms and tight error bounds". In: 2014 IEEE 55th annual symposium on foundations of computer science. IEEE. 2014, pp. 464–473 (cit. on p. 262).

[Car+20] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. "Extracting training data from large language models". In: arXiv preprint arXiv:2012.07805 (2020) (cit. on p. 257).

[CHBB20] J. P. Cohen, M. Hashir, R. Brooks, and H. Bertrand. "On the limits of cross-domain generalization in automated X-ray prediction". In: Medical Imaging with Deep Learning. PMLR. 2020, pp. 136–155 (cit. on p. 252).

[Che+17] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models". In: Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017, pp. 15–26 (cit. on p. 256).

[Cho+21] C. A. Choquette-Choo, N. Dullerud, A. Dziedzic, Y. Zhang, S. Jha, N. Papernot, and X. Wang. "CaPC Learning: Confidential and Private Collaborative Learning". In: arXiv preprint arXiv:2102.05188 (2021) (cit. on p. 273).

[CMS11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. "Differentially private empirical risk minimization". In: Journal of Machine Learning Research 12.Mar (2011), pp. 1069–1109 (cit. on pp. 259, 262).

[CTCP20] C. A. C. Choo, F. Tramer, N. Carlini, and N. Papernot. "Label-only membership inference attacks". In: arXiv preprint arXiv:2007.14321 (2020) (cit. on p. 255).

[CY15] Y. Cao and J. Yang. "Towards making systems forget with machine unlearning". In: 2015 IEEE Symposium on Security and Privacy. IEEE. 2015, pp. 463–480 (cit. on p. 272).

[De+22] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle. "Unlocking high-accuracy differentially private image classification through scale". In: arXiv preprint arXiv:2204.13650 (2022) (cit. on pp. 271, 272).

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating noise to sensitivity in private data analysis". In: Theory of cryptography conference. Springer. 2006, pp. 265–284 (cit. on p. 254).

[DR+14] C. Dwork, A. Roth, et al. "The algorithmic foundations of differential privacy." In: Foundations and Trends in Theoretical Computer Science 9.3-4 (2014), pp. 211–407 (cit. on pp. 267, 268).

[Dwo+15] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. "Generalization in adaptive data analysis and holdout reuse". In: arXiv preprint arXiv:1506.02629 (2015) (cit. on p. 270).

[EMRS19] Ú. Erlingsson, I. Mironov, A. Raghunathan, and S. Song. "That which we call private". In: arXiv preprint arXiv:1908.03566 (2019) (cit. on p. 276).

[GBCB16] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. Deep learning. Vol. 1. MIT Press, 2016 (cit. on p. 252).

[Gil+16] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy". In: International conference on machine learning. PMLR. 2016, pp. 201–210 (cit. on p. 273).

[Irv+19] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison". In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. 01. 2019, pp. 590–597 (cit. on p. 252).

[IS15] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: International conference on machine learning. PMLR. 2015, pp. 448–456 (cit. on p. 266).

[Iye+19] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. "Towards practical differentially private convex optimization". In: 2019 IEEE Symposium on Security and Privacy (SP). 2019 (cit. on p. 259).

[Jia+19] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong. "Memguard: Defending against black-box membership inference attacks via adversarial examples". In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019, pp. 259–274 (cit. on pp. 258, 278).

[JUO20] M. Jagielski, J. Ullman, and A. Oprea. "Auditing differentially private machine learning: How private is private sgd?" In: arXiv preprint arXiv:2006.07709 (2020) (cit. on p. 276).

[KB14] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: arXiv preprint arXiv:1412.6980 (2014) (cit. on p. 265).

[Kon+16] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. "Federated learning: Strategies for improving communication efficiency". In: arXiv preprint arXiv:1610.05492 (2016) (cit. on p. 273).

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: Advances in neural information processing systems 25 (2012), pp. 1097–1105 (cit. on p. 252).

[KSL19] S. Kornblith, J. Shlens, and Q. V. Le. "Do better imagenet models transfer better?" In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 2661–2671 (cit. on p. 275).

[KST12] D. Kifer, A. Smith, and A. Thakurta. "Private convex empirical risk minimization and high-dimensional regression". In: Conference on Learning Theory. 2012, pp. 25–1 (cit. on p. 259).

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: nature 521.7553 (2015), pp. 436–444 (cit. on pp. 251, 252).

[LF20] K. Leino and M. Fredrikson. "Stolen memories: Leveraging model memorization for calibrated white-box membership inference". In: 29th {USENIX} Security Symposium ({USENIX} Security 20). 2020, pp. 1605–1622 (cit. on p. 258).

[Lon+18] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen. "Understanding membership inferences on well-generalized learning models". In: arXiv preprint arXiv:1802.04889 (2018) (cit. on p. 258).

[Low99] D. G. Lowe. "Object recognition from local scale-invariant features". In: Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. Ieee. 1999, pp. 1150–1157 (cit. on p. 252).

[LTLH21] X. Li, F. Tramer, P. Liang, and T. Hashimoto. "Large language models can be strong differentially private learners". In: arXiv preprint arXiv:2110.05679 (2021) (cit. on p. 272).

[Man13] A. Mantelero. "The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'". In: Computer Law & Security Review 29.3 (2013), pp. 229–235 (cit. on p. 272).

[Mir17] I. Mironov. "Rényi differential privacy". In: 2017 IEEE 30th computer security foundations symposium (CSF). IEEE. 2017, pp. 263–275 (cit. on p. 264).

[MRTZ17] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. "Learning differentially private recurrent language models". In: arXiv preprint arXiv:1710.06963 (2017) (cit. on p. 275).

[MSDS19] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. "Exploiting unintended feature leakage in collaborative learning". In: 2019 IEEE Symposium on Security and Privacy (SP). 2019, pp. 691–706 (cit. on p. 273).

[MZH19] Y. Ma, X. Zhu, and J. Hsu. "Data poisoning against differentially-private learners: Attacks and defenses". In: arXiv preprint arXiv:1903.09860 (2019) (cit. on p. 277).

[Nas+21] M. Nasr, S. Song, A. Thakurta, N. Papernot, and N. Carlini. "Adversary instantiation: Lower bounds for differentially private machine learning". In: arXiv preprint arXiv:2101.04535 (2021) (cit. on p. 277).

[Nee+15] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens. "Adding gradient noise improves learning for very deep networks". In: arXiv preprint arXiv:1511.06807 (2015) (cit. on p. 264).

[NRS07] K. Nissim, S. Raskhodnikova, and A. Smith. "Smooth sensitivity and sampling in private data analysis". In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. 2007, pp. 75–84 (cit. on p. 269).

[NS08] A. Narayanan and V. Shmatikov. "Robust de-anonymization of large sparse datasets". In: 2008 IEEE Symposium on Security and Privacy (sp 2008). IEEE. 2008, pp. 111–125 (cit. on p. 259).

[NSH18] M. Nasr, R. Shokri, and A. Houmansadr. "Machine Learning with Membership Privacy using Adversarial Regularization". In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018, pp. 634–646 (cit. on pp. 258, 278).

[OM15] E. Oyallon and S. Mallat. "Deep roto-translation scattering for object classification". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 2865–2873 (cit. on p. 274).

[Pap+16] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. "Semi-supervised knowledge transfer for deep learning from private training data". In: arXiv preprint arXiv:1610.05755 (2016) (cit. on pp. 266, 269).

[Pap+17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. "Practical black-box attacks against machine learning". In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. 2017, pp. 506–519 (cit. on pp. 256, 266).

[Pap+20] N. Papernot, A. Thakurta, S. Song, S. Chien, and Ú. Erlingsson. "Tempered sigmoid activations for deep learning with differential privacy". In: arXiv preprint arXiv:2007.14191 (2020) (cit. on pp. 271, 275).

[PS21] N. Papernot and T. Steinke. “Hyperparameter tuning with renyi differential privacy”. In: arXiv preprint arXiv:2110.03620 (2021) (cit. on p. 275).

[Rad+19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. “Language models are unsupervised multitask learners”. In: OpenAI blog 1.8 (2019), p. 9 (cit. on p. 257).

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: nature 323.6088 (1986), pp. 533–536 (cit. on p. 263).

[Sal+16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. “Improved techniques for training gans”. In: Advances in neural information processing systems 29 (2016), pp. 2234–2242 (cit. on p. 270).

[Sal+18] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. “Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models”. In: arXiv preprint arXiv:1806.01246 (2018) (cit. on p. 255).

[SCS13] S. Song, K. Chaudhuri, and A. D. Sarwate. “Stochastic gradient descent with differentially private updates”. In: 2013 IEEE Global Conference on Signal and Information Processing. IEEE. 2013, pp. 245–248 (cit. on p. 262).

[Sha+18] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Dahl. “Measuring the effects of data parallelism on neural network training”. In: arXiv preprint arXiv:1811.03600 (2018) (cit. on p. 262).

[SPGG21] V. M. Suriyakumar, N. Papernot, A. Goldenberg, and M. Ghassemi. “Chasing Your Long Tails: Differentially Private Prediction in Health Care Settings”. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. 2021, pp. 723–734 (cit. on p. 270).

[SSM19] L. Song, R. Shokri, and P. Mittal. “Privacy risks of securing machine learning models against adversarial examples”. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019, pp. 241–257 (cit. on p. 277).

[SSSS17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership inference attacks against machine learning models". In: 2017 IEEE symposium on security and privacy (SP). IEEE. 2017, pp. 3–18 (cit. on pp. 253, 254, 258).

[STT20] S. Song, O. Thakkar, and A. Thakurta. "Characterizing Private Clipped Gradient Descent on Convex Generalized Linear Problems". In: arXiv preprint arXiv:2006.06783 (2020) (cit. on p. 271).

[Sze+13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: arXiv preprint arXiv:1312.6199 (2013) (cit. on pp. 256, 277).

[TB18] F. Tramer and D. Boneh. "Slalom: Fast, verifiable and private execution of neural networks in trusted hardware". In: arXiv preprint arXiv:1806.03287 (2018) (cit. on p. 273).

[TB20] F. Tramèr and D. Boneh. "Differentially Private Learning Needs Better Features (or Much More Data)". In: arXiv preprint arXiv:2011.11660 (2020) (cit. on p. 274).

[YGFJ18] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. "Privacy risk in machine learning: Analyzing the connection to overfitting". In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF). 2018, pp. 268–282 (cit. on pp. 255–258).

[Zha+16] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". In: arXiv preprint arXiv:1611.03530 (2016) (cit. on p. 263).

[Zha+19] Z. Zhao, N. Papernot, S. Singh, N. Polyzotis, and A. Odena. "Improving differentially private models with active learning". In: arXiv preprint arXiv:1910.01177 (2019) (cit. on p. 272).