Interpretable and Differentially Private Predictions

Frederik Harder,[1,2] **Matthias Bauer,**[1,3*] **Mijung Park**[1,2]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Department of Computer Science, University of Tübingen, Tübingen, Germany
[3]Department of Engineering, University of Cambridge, Cambridge, UK
{fharder, bauer, mpark}@tue.mpg.de

Abstract

Interpretable predictions, which clarify why a machine learning model makes a particular decision, can compromise privacy by revealing the characteristics of individual data points. This raises the central question addressed in this paper: *Can models be interpretable without compromising privacy?* For complex "big" data fit by correspondingly rich models, balancing privacy and explainability is particularly challenging, such that this question has remained largely unexplored. In this paper, we propose a family of simple models with the aim of approximating complex models using several *locally linear maps* per class to provide high classification accuracy, as well as differentially private explanations on the classification. We illustrate the usefulness of our approach on several image benchmark datasets as well as a medical dataset.

1 Introduction

The *General Data Protection Regulation (GDPR)* by the European Union imposes two important requirements on algorithmic design, *interpretability* and *privacy* (Voigt and Bussche 2017). These requirements introduce new standards on future algorithmic techniques, making them of particular concern to the machine learning community (Goodman and Flaxman 2016). This paper addresses these two requirements in the context of classification, and studies the trade-off between privacy, accuracy and interpretability, see Fig. 1.

Broadly speaking, there are two options to take for gaining interpretability: (i) rely on *inherently interpretable models*; and (ii) rely on *post-processing schemes* to probe trained complex models. Inherently interpretable models are often relatively simple and their predictions can be easily analyzed in terms of their respective input features. For instance, in logistic regression classifiers and sparse linear models the coefficients represent the importance of each input feature. However, modern "big" data typically exhibit complex patterns, such that these relatively simplistic models often have lower accuracy than more complex ones.
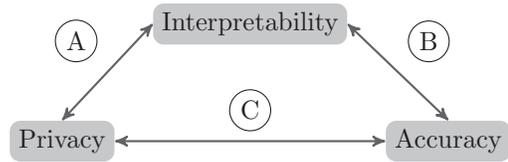
*Matthias Bauer is now at DeepMind

Figure 1: Modern machine learning systems need to trade off accuracy, privacy, and interpretability.

In order to soften this trade-off between interpretability and accuracy (Fig. 1 Ⓑ), many post-processing schemes aim to gain insights from complex models like deep neural networks. One prominent aspect of these approaches is the use of gradient-based attributions (Selvaraju et al. 2016; Ribeiro, Singh, and Guestrin 2016; Smilkov et al. 2017; Sundararajan, Taly, and Yan 2017; Montavon et al. 2015; Bach et al. 2015; Ancona et al. 2017).

In another line of research, many recent papers address the concern that complex models with outstanding predictive performance can expose sensitive information from the dataset they were trained on (Carlini et al. 2018; Song, Ristenpart, and Shmatikov 2017; Shokri and Shmatikov 2015; Fredrikson, Jha, and Ristenpart 2015). To quantify privacy, several approaches adopt the notion of *differential privacy* (DP), which provides a mathematically provable definition of privacy, and can quantify the level of privacy an algorithm or a model provides (Dwork and Roth 2014). In plain English, an algorithm is called differentially private (DP), if its output is random enough to obscure the participation of any single individual in the data. The randomness is typically achieved by injecting noise into the algorithm. The amount of noise is determined by the level of privacy the algorithm guarantees and the sensitivity, a maximum difference in its output depending on a single individual's participation or non-participation in the data (see Sec. 2 for the mathematical definition of DP).

There is, however, a natural trade-off between privacy and accuracy (Fig. 1 Ⓒ): A large amount of added noise provides a high level of privacy but also harms prediction accuracy. When the number of parameters is high, like in deep neural network models, juggling this trade-off is very

challenging, as privatizing high dimensional parameters results in a high privacy loss to meet a good prediction level (Abadi et al. 2016). Therefore, most existing work in differential privacy literature considers relatively small networks or assumes that some relevant data are publicly available to train a significant part of the network without privacy violation to deal with the trade-off (see Sec. 5 for details). However, none of the existing work takes into account the interpretability of the learned models and this is our core contribution described below.

Our contribution. In this paper, we study the trade-off between interpretability, privacy, and accuracy by making the following three contributions.

- **We propose a novel family of interpretable models**: To take into account privacy *and* interpretability (Fig. 1 (A)), we propose a family of inherently interpretable models that can be trained privately. These models approximate the mapping of a complex model from the input data to class score functions, using several *locally linear maps* (LLM) per class. Our formulation for LLM is inspired by the approximation of differentiable functions as a collection of piece-wise linear functions, i.e., the first-order Taylor expansions of the function at a sufficiently large number of input locations. With an adequate number of linear maps, our local models permit a relatively slight loss in accuracy compared to complex model counterparts. The level of loss in prediction accuracy depends on the complexity of data.

- **We provide DP "local" and "global" explanations on classification**: Our model LLM, trained with the DP constraint, provides insights on the key features for classification at a "local" and "global" level. A *local* explanation of a model illustrates how the model behaves at and around a specific input, showing how relevant different features of the input were to the model decision. This is a typical outcome one could obtain by probing a complex model using existing attribution methods. However, our model also provides a *global* explanation, illustrating how the model functions as a whole and, in the case of classification, what types of input the different classes are sensitive to. This is what distinguishes our work from other existing post-processing attribution methods.

- **We propose to use random projections to better deal with privacy and accuracy trade-off**: We propose to adopt the *Johnson-Lindenstrauss transform*, a.k.a., *random projection* (Kenthapadi et al. 2013), to decrease the dimensionality of each LLM and then privatize the resulting lower dimensional quantities. We found that exploiting data-indepdent random projection achieves a significantly better trade-off for high-dimensional image data.

We would like to emphasize that our work is the first to address the interplay between interpretability, privacy, and accuracy. Hence, this work presents not only a novel inherently interpretable model but also an important conceptual contribution to the field that will spur more research on this intersection.

2 Background on Differential Privacy

We start by introducing differential privacy and a composition method that we will use in our algorithm, as well as random projections.

Differential privacy. Consider an algorithm $\mathcal{M}$ and neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing by a single entry, where the dataset $\mathcal{D}'$ is obtained by excluding one datapoint from the dataset $\mathcal{D}$. In DP (Dwork and Roth 2014), the quantity of interest is *privacy loss*, defined by

$$L^{(o)} = \log \frac{\Pr(\mathcal{M}_{(\mathcal{D})} = o)}{\Pr(\mathcal{M}_{(\mathcal{D}')} = o)}, \qquad (1)$$

where $\mathcal{M}_{(\mathcal{D})}$ and $\mathcal{M}_{(\mathcal{D}')}$ denote the outputs of the algorithm given $\mathcal{D}$ and $\mathcal{D}'$, respectively. $\Pr(\mathcal{M}_{(\mathcal{D})} = o)$ denotes the probability that $\mathcal{M}$ returns a specific output o. When the two probabilities in Eq. (1) are similar, even a strong adversary, who knows all the datapoints in $\mathcal{D}$ except for one, could not discern the one datapoint by which $\mathcal{D}$ and $\mathcal{D}'$ differ, based on the output of the algorithm alone. On the other hand, when the probabilities are very different, it would be easy to identify the exclusion of the single datapoint in $\mathcal{D}'$. Hence, the privacy loss quantifies how revealing an algorithm's output is about a single entry's presence in the dataset $\mathcal{D}$. Formally, an algorithm $\mathcal{M}$ is called ϵ-DP if and only if $|L^{(o)}| \leq \epsilon, \forall o$ and for all neighbouring datasets $\mathcal{D}, \mathcal{D}'$. A weaker version of the above is (ϵ, δ)-DP, if and only if $|L^{(o)}| \leq \epsilon$, with probability at least $1 - \delta$.

Introducing a noise addition step is a popular way of making an algorithm DP. The *output perturbation* method achieves this by adding noise to the output h, where the noise is calibrated to h's *sensitivity*, defined by

$$S_h = \max_{\mathcal{D}, \mathcal{D}', |\mathcal{D}-\mathcal{D}'|=1} \|h(\mathcal{D}) - h(\mathcal{D})\|_2, \qquad (2)$$

which is the maximum difference in terms of L2-norm, under the one datapoint's difference in $\mathcal{D}$ and $\mathcal{D}'$. With the sensitivity, we can privatize the output using the *Gaussian mechanism*, which simply adds Gaussian noise of the form: $\tilde{h}(\mathcal{D}) = h(\mathcal{D}) + \mathcal{N}(0, S_h^2\sigma^2\mathbf{I}_p)$, where $\mathcal{N}(0, S_h^2\sigma^2\mathbf{I}_p)$ means the Gaussian distribution with mean 0 and covariance $S_h^2\sigma^2\mathbf{I}_p$. The resulting quantity $\tilde{h}(\mathcal{D})$ is (ϵ, δ)-DP, where $\sigma \geq \sqrt{2\log(1.25/\delta)}/\epsilon$ (see Dwork and Roth (2014) for a proof). In this paper, we use the Gaussian mechanism to achieve differentially private LLM.

Properties of differential privacy. DP has two important properties: (i) post-processing invariance and (ii) composability. *Post-processing invariance* states that applying any *data-independent* mechanism to a DP quantity does not alter the privacy level of the resulting quantity.

Composability states that combining DP quantities degrades privacy in a quantifiable way. The most naïve way is the *linear* composition (Theorem 3.14 in Dwork and Roth (2014)), where the resulting parameter, which is often called *cumulative privacy loss* (cumulative ϵ and δ), are linearly summed up, $\epsilon = \sum_{t=1}^{T} \epsilon_t$ and $\delta = \sum_{t=1}^{T} \delta_t$ after the repeated use of data T times with the per-use privacy loss

(ϵ_t, δ_t). Recently, Abadi et al. (2016) proposed the *moments accountant* method, which provides an efficient way of combining ϵ and δ such that the cumulative privacy loss is significantly smaller than that by other composition methods. The resulting composition provides a better utility, meaning that a smaller amount of noise is required to add for the same privacy guarantee compared to other composition methods.

Random projections in the context of differential privacy
A variant of our method involves projecting each input onto a lower-dimensional space using a *Johnson-Lindenstrauss transform* (a.k.a., *random projection*) (Kenthapadi et al. 2013). We construct the projection matrix $\mathbf{R}$ by drawing each entry from $\mathcal{N}(0, 1/D')$ where D' is the dimension of the projected space. This projection nearly preserves the distances between two points in the data space and in the embedding space, as this projection guarantees low-distortion embeddings. Random projections have been used to ensure DP (Blocki et al. 2012). However, here we only utilize them as a convenient method to reduce input dimension to our learnable linear maps. Since the random filters are data-independent, they do not need to be privatized.

3 Our method: Locally Linear Maps (LLM)

Motivation. As mentioned earlier, complex models such as deep neural networks tend to lack interpretability due to their nested feature structure. Gradient-based attribution methods can provide local explanations by computing a linear approximation of the model at a given point in the input space (see Sec. 5 for more details). Such approximations can be seen as sensitivity maps that highlight which parts of the input affect the model decision locally. However, these approaches lack *global* explanations that provide insight on how the model works as a whole, e.g., it is not straightforward to obtain class-wise key features. Furthermore, existing methods in the DP literature do not take into account the interpretability of learned models. In order to satisfy both interpretability and privacy demands, we desire a model with the following properties:

1. It can provide both *local and global explanations*.

2. It has efficient ways to limit in the number of parameters to achieve a *good privacy accuracy trade-off*.

3. It is *more expressive than standard linear models* to capture complex patterns in the data.

Locally Linear Maps (LLM). We introduce a set of local functions f_k for each class k, and parameterize each f_k by a combination of M linear maps denoted by g_m^k. The M linear maps are weighted separately for each class using the weighting coefficients $\sigma(\mathbf{x})_m^k$, which determine how *important* each linear map is for classifying a given input:

$$f_k(\mathbf{x}) = \sum_{m=1}^{M} \sigma(\mathbf{x})_m^k \, g_m^k(\mathbf{x}), \text{ s.t. } g_m^k(\mathbf{x}) = {\mathbf{w}_m^k}^{\top}\mathbf{x} + \mathbf{b}_m^k,$$

and $\sigma_m^k(\mathbf{x}) = \dfrac{\exp\left[\beta \cdot g_m^k(\mathbf{x})\right]}{\sum_{m=1}^{M} \exp\left[\beta \cdot g_m^k(\mathbf{x})\right]}.$ (3)

One way to choose the weighting coefficients is by assigning a probability to each linear map using the softmax function as in Eq. (3). We introduce a global inverse temperature parameter β in the softmax to tune the sensitivity of the relative weighting – large β (small temperature) favors single filters; small β (high temperature) favors several filters. The softmax weighting is useful for avoiding non-identifiability issues of parameters in mixture models. More importantly, the softmax weighting assigns an importance to each map particular to this example. In other words, it provides rankings of filters for different examples even if they are classified as the same class. We revisit this point in Sec. 4. We train the LLM by optimizing the following (standard) cross-entropy loss:

$$\mathcal{L}(\mathbf{W}, \mathcal{D}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{n,k} \log \hat{y}_{n,k}(\mathbf{W}),$$ (4)

where we denote the parameters of LLM collectively by $\mathbf{W}$, and we define the predictive class label by the mapping from the pre-activation through another softmax function.

$$\hat{y}_{n,k}(\mathbf{W}) = \exp(f_k(\mathbf{x}_n))/[\sum_{k'=1}^{K}\exp(f_{k'}(\mathbf{x}_n))]$$ (5)

When the number of filters per class is one, this reduces to logistic regression; increasing the number of filters adds expressive capacity to each class. The classification is approximately linear at the location of the input, which means that locally each model decision from a certain input can be explained using only the active filters, as we illustrate in the remainder of the paper. In addition the shallow nature of the model lends itself to global interpretability, as the filter-bank for each class is easily accessible and provides an overview of the inputs this class is sensitive to.

LLM as neural network approximations. One interpretation of LLM is linearizations of neural networks. Suppose we trained a neural network model on a K-class classification problem, where the network maps a high dimensional input $\mathbf{x} \in \mathbb{R}^D$ to a class score function $\mathbf{s}(\mathbf{x})$, i.e., the pre-activation before the final softmax, where $\mathbf{s}(\mathbf{x})$ is a K-dimensional vector with entries s_k. Denote the mapping $\phi : \mathbf{x} \mapsto \mathbf{s}(\mathbf{x})$ and the parameters of the network by $\boldsymbol{\theta}$. We would like to find the best approximation to the function ϕ, which presents interpretable features for classification and also guarantees a certain level of privacy. For this, we take inspiration from gradient-based attribution methods for deep neural networks (Ancona et al. 2017). These methods assume a set of attributions, at which the gradients of a classifier with respect to the input are maximized, *and* that the gradient information provides interpretability as to why the classifier makes a certain prediction. More specifically, these consider a first order Taylor approximation of ϕ,

$$\boldsymbol{\phi}(\mathbf{x}) \approx \boldsymbol{\phi}(\mathbf{x}_0) + \boldsymbol{\phi}'(\mathbf{x}_0)^{\top}(\mathbf{x} - \mathbf{x}_0) = \boldsymbol{\phi}'(\mathbf{x}_0)^{\top}\mathbf{x} + \boldsymbol{\alpha},$$

where $\boldsymbol{\phi}'(\mathbf{x}_0) = \left[\frac{\partial}{\partial \mathbf{x}}\boldsymbol{\phi}(\mathbf{x})\right]_{\mathbf{x}=\mathbf{x}_0}$, and shift term $\boldsymbol{\alpha} = \boldsymbol{\phi}(\mathbf{x}_0) - \boldsymbol{\phi}'(\mathbf{x}_0)^{\top}\mathbf{x}_0$. Therefore, finding *good* input locations

$\mathbf{x}_0$ to make the first order approximation to the function ϕ and using their gradient information would reveal the discriminative features of a given classifier.

There are two problems in directly using this approach. First, it is challenging to identify which input points (and how many of them) are *informative* to make an interpretable linear approximation of the classifier. Second, directly using ϕ and its gradients violates privacy, as ϕ contains sensitive information about individuals from the training dataset. Privatizing ϕ requires computing the sensitivity, which determines an appropriate amount of noise to add (see Sec. 2). In case of deep neural network models, we cannot analytically identify one datapoint's contribution to the learned function ϕ appeared in Eq. (2). Thus, we cannot use the raw function ϕ and its gradients, unless we privatize the parameters of ϕ[1]. For these reasons, extracting a private approximation of ϕ is difficult and we instead opt to train a model of the same form from scratch, leading us to LLMs, as described above.

3.1 Differentially private LLM

To produce differentially private LLM parameters $\widetilde{\mathbf{W}}$, we adopt the moments accountant method combined with the gradient-perturbation technique, called, *differentially private stochastic gradient descent* (DP-SGD) (Abadi et al. 2016). In our work, we perturb gradients at each learning step when optimizing Eq. (4) for all LLM parameters $\mathbf{W}$; and compute the cumulative privacy loss using the moments accountant method.

When we perturb the gradient, we must ensure to add the right amount of noise. As there is no way of knowing how much change a single datapoint would make in the gradient's L2-norm, we rescale all the datapoint-wise gradients, $\mathbf{h}_t(\mathbf{x}_n) := \nabla_{\mathbf{W}} \mathcal{L}(\mathbf{W}, \mathcal{D}_n)$ for all $n = \{1, \cdots, N\}$, by a pre-defined norm clipping threshold, C, as used in (Abadi et al. 2016), i.e., $\bar{\mathbf{h}}_t(\mathbf{x}_n) \leftarrow \mathbf{h}_t(\mathbf{x}_n)/\max(1, \|\mathbf{h}_t(\mathbf{x}_i)\|_2/C)$. Algorithm 1 summarizes this procedure. We formally state that the resulting LLM is DP in Theorem 3.1.

Algorithm 1 DP-LLM for interpretable classification

Require: Dataset $\mathcal{D}$, norm-clipping threshold C, privacy parameter σ^2, and learning rate η_t

Ensure: (ϵ, δ)-DP locally linear maps for all K classes, $\widetilde{\mathbf{W}}$

for number of training steps $t \leq T$ **do**

    1: For each minibatch of size L, we noise up the gradient after clipping the norm of the datapoint-wise gradient via $\tilde{\mathbf{h}}_t \leftarrow \frac{1}{L} \left[\sum_{n=1}^{L} \bar{\mathbf{h}}_t(\mathbf{x}_n) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \right]$.

    2: Then, we make a step in the descending direction by $\widetilde{\mathbf{W}}_{t+1} \leftarrow \widetilde{\mathbf{W}}_t - \eta_t \tilde{\mathbf{h}}_t$.

end for

Calculate the cumulative privacy loss (ϵ, δ) using the moments accountant.

Theorem 3.1. *Algorithm 1 produces (ϵ, δ)-DP locally linear maps for all K classes.*

---

[1]Once ϕ is privatized, we can safely use post-processing methods for interpretability. A comparison to this is shown in Sec. 4.

Proof: Given an initial *data-independent* value of $\mathbf{W}_0$, if we add Gaussian noise to the norm-clipped gradient evaluated on the subsampled data with the sampling rate $q = L/N$, then due to the *Gaussian mechanism* (Theorem 3.22 in (Dwork and Roth 2014)), the resulting estimate $\tilde{\mathbf{W}}_1$ from a single gradient step (i.e., the step **2** in Algorithm 1) is (ϵ', δ')-DP, where $\sigma \geq c \cdot q \sqrt{\log(1/\delta')}/\epsilon'$ with some constant c. The cumulative privacy loss after T repetitions of step **2** in Algorithm 1 is (ϵ, δ)-DP, which is a direct consequence of Theorem 1 in (Abadi et al. 2016).

Improving privacy and accuracy trade-off under LLM
For high-dimensional inputs such as images, we found that adding noise to the gradient corresponding to the full dimension of $\mathbf{W}$ led to very low accuracies for private training. Therefore, we propose to incorporate the random projection matrix $\mathbf{R}_m \in \mathbb{R}^{D' \times D}$ with $D' \ll D$, which is shared among all classes k, to first decrease the dimensionality of the parameters that need to be privatized. Each LLM is therefore parameterized as $\mathbf{w}_m^k = \mathbf{m}_m^k \mathbf{R}_m$, where the effective parameter for each local linear map is $\mathbf{m}_m^k \in \mathbb{R}^{D'}$. We perturb the gradient of $\mathbf{m}_m^k$ for all k and m in each training step in Algorithm 1 to produce DP linear maps, $\widetilde{\mathbf{w}}_m^k = \widetilde{\mathbf{m}}_m^k \mathbf{R}_m$.

Due to the post-processing invariance property, we can use the differentially private LLM to make predictions on test data. Here we focus on guarding the training data's privacy and assume that the test data do not need to be privatized, which is a common assumption in DP literature.

4 Experiments

In this section we evaluate the trade-off between accuracy, privacy, and interpretability for our LLM model on several datasets and compare to other methods where it is appropriate. Our implementation is available on GitHub[2].

4.1 MNIST Classification

Problem. We consider the classification of MNIST (LeCun and Cortes 2010) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) images with the usual train/test splits and train a CNN[3] as a baseline model, which has two convolutional layers with 5x5 filters and first 20, then 50 channels each followed by max-pooling and finally a fully connected layer with 500 units. The model achieves 99% test accuracy on MNIST and 87% on Fashion-MNIST.

Setup. We train several LLMs in the private and non-private setting. By default, we use LLM models with $M = 30$ filters per class and random projections to $D' = 300$ dimensions, which are optimized for 20 epochs using the Adam optimizer with learning rate 0.001, decreasing by 20% every 5 epochs. On MNIST the model benefits from a decreased inverse softmax temperature $\beta = 1/30$, while $\beta = 1$ is optimal for Fashion-MNIST. We choose a large

---

[2]github.com/frhrdr/dp-llm
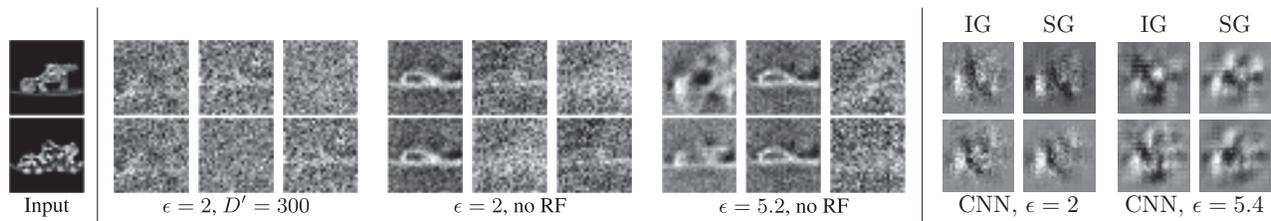
[3]github.com/pytorch/examples/blob/master/mnist/main.py

Figure 2: For 2 test inputs (left) and 3 different DP-LLM setups (groups of 3 columns), we show the 3 highest activated filters in desending order. We look at the default setting with random filters at $D' = 300$ and $\epsilon = 2$ (center left), the same setting without random filters (center), and at a lower privacy $\epsilon = 5.2$ setting (center right). Attribution plots from DP-CNNs at matching privacy levels on the same input are shown for comparison (right).

batch size of 500, as this improves the signal-to-noise ratio of our algorithm. In the private setting we clip the per-sample gradient norm to $C = 0.001$ and train with $\sigma = 1.3$, which gives this model an $(\epsilon = 2, \delta = 10^{-5})$-DP guarantee via the moments accountant. For the low privacy regime $\epsilon \geq 4$ we train with a batch size of 1500 and for 60 epochs.
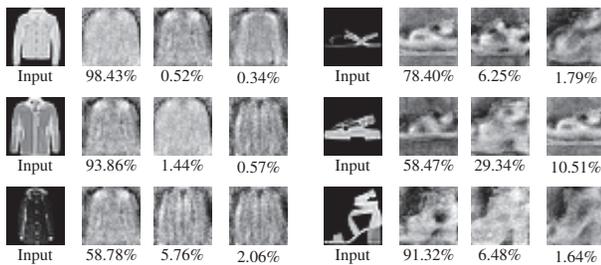
Figure 3: Top 3 filters with associated weightings for test images from two classes. In most cases, a single filter dominates the softmax selection for the class.
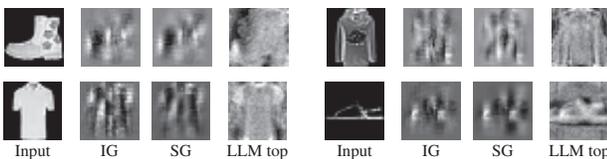

Figure 4: Comparison of CNN and LLM interpretability. *left to right*: input image, integrated gradient (IG) of CNN, smoothed gradient (SG) of CNN, and the top filter of LLM. The CNN attribution match well but aren't as easy to interpret as the simple filter.

Inherent interpretability. In order to highlight the interpretability of the LLM architecture, we compare learned filters of our model to two attribution methods applied to a neural network trained on the same data. We train a simple CNN and an LLM on Fashion-MNIST to matching 87% test accuracy and then visualize the CNN's sensitivity to test images using SmoothGrad (SG) (Smilkov et al. 2017) and integrated gradients (IG) (Sundararajan, Taly, and Yan 2017)

and compare these methods to LLM filters in Fig. 4. Note that we do not multiply the integrated gradient with the input image, as Fashion-MNIST images have a mask-like effect which occludes the partial output of the method. We observe that both alternative attribution methods produce similar outputs, which are nonetheless hard to interpret, whereas the LLM filters show simplistic prototype images of the corresponding classes. This is further illustrated in Fig. 3 where we show the three highest weighted filters for test images from three classes. The diversity of filters varies for different class labels, as some are more varied and harder to discriminate than others. For instance, while the sandal class (right) has filters which distinguish between different types of heels, the coat filters (left) are mostly selective in the shoulder region and general silhouette, which is sufficient for classifying a majority of the inputs correctly, but some filters also track other features like arms, collar and zipper. The relevance weights for each filter show that in most cases, the top filter is assigned almost all the weight, indicating that the softmax is a good approximation of the maximum and the class features are indeed approximately linear locally.

Trade-offs with interpretability (Fig. 1 Ⓐ and Ⓑ). We investigate the learned LLM filters under increasing privacy guarantees and increased private utility as shown in Fig. 2. For two test inputs we plot the filters with highest activation in three DP setups. We compare the default setting with random filters at $D' = 300$ and $\epsilon = 2$, the same setting without random filters, and a lower privacy setting trained with $\epsilon = 5.2$. The default setting optimizes privacy and utility at the expense of interpretability. As the figure shows, removing the random projections and reducing the level of privacy gradually restores interpretability of the filters. On the very right, we show attribution plots from CNNs trained with DP-SGD at $\epsilon = 2$ and $\epsilon = 5.4$ on the same input for reference. When comparing to Fig. 4, one can see that the quality of the CNN attributions is diminished by the privacy constraints as well, to the point where it is hard to make out any connection to the input image.

Privacy vs. accuracy. (Fig. 1 Ⓒ) In Fig. 5 (left) we show the trade-off of privacy strength and accuracy in our model. Note that current privatized network methods (Abadi
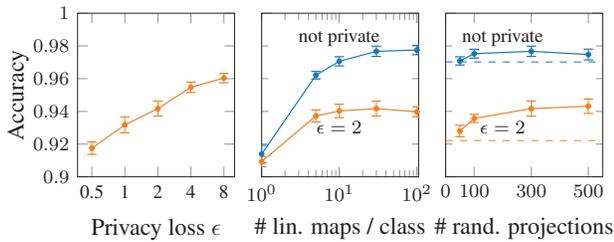
Figure 5: Accuracy of our LLM model on the MNIST testset for different levels of privacy and different model configurations in the private (orange) and non-private (blue) setting. Errorbars are 2 stdev from 10 random restarts; dashed lines on the right denote no random projections.

et al. 2016; Phan, Wu, and Dou 2017) achieve an accuracy of 95% for $\epsilon = 2$ and up to 92% for $\epsilon = 0.5$, which is comparable to our mean accuracy of $94.2 \pm 0.4\%$ and $91.8 \pm 0.4\%$ respectively (on Fashion-MNIST we achieve $80.7 \pm 0.6\%$ and $83.2 \pm 0.4\%$). However, such a privatized network does not provide transparent explanations as opposed to our approach. Another popular reference model is the PATE method (Papernot et al. 2018), which trains a student model to 98% at $\epsilon = 2$ on MNIST using an ensemble of teachers and additional public data. We do not consider this setup here, as the accuracy of the teacher votes alone lies at 94.4% in the nonprivate setting, highlighting the importance of the additional data. In the remainder of Fig. 5 we study the impact of varying the number of filters per class M (center) and the output dimensionality of the random projections D' (right) in private and non-private LLM models. Private LLMs deteriorate beyond a certain number of linear maps due to the increased noise needed to privatize them, whereas non-private models continue to benefit from additional filters. Increasing the dimensionality of the random projections benefits private training.

4.2 Disease classification in a medical dataset

Problem. As a second task we consider disease classification in the Henan Renmin Hospital Data (Li et al. 2017; Maxwell et al. 2017)[4]. It contains 110,300 medical records with 62 input features and 3 binary outputs. The input features are 4 basic examinations (sex, BMI, distolic, systolic), 26 items from blood examinations, 12 items from urine examinations, and 20 items from liver function tests. The three binary outputs denote three medical conditions – hypertension, diabetes, and fatty liver – which can also co-occur. Following (Maxwell et al. 2017) we transform this multi-label task into a multi-class problem by considering the powerset of the three binary choices as eight independent classes. Because these classes are highly imbalanced, we only retain the four most common classes, leaving us with 100,140 records.

Setup. By default, we use an LLM model with $M = 2$ filters per class and no random projections, which is optimized

---

[4]downloaded from http://pinfish.cs.usm.edu/dnn/

for 20 epochs using the Adam optimizer with learning rate 0.01, decreasing by 20% every 5 epochs. We choose a batch size of 256. In the private setting we clip the per-sample gradient norm to 0.001 and train with $\sigma = 1.25$, which gives this model an $(\epsilon \approx 1.5, \delta = 2 \cdot 10^{-5})$-DP guarantee.
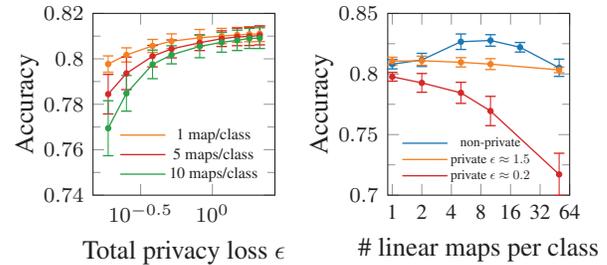

Figure 6: Accuracy of our LLM model on the Henan Renmin Hospital testset for different levels of privacy and different model configurations in the private and non-private setting. Errorbars are 2 stdev for 10 random restarts.

We train a baseline DNN (3 dense hidden layers with 128 units each) as well as several LLMs with varying number of linear filters per class in private and non-private settings. In Fig. 6 we visualize the trade-off between accuracy and privacy for varying privacy losses as well as numbers of linear maps. Like before, the accuracy deteriorates as we decrease the privacy loss (Fig. 6 *top*). As the number of linear maps per class is increased (Fig. 6 *bottom*), the accuracy for the private models also drops due to the privacy budget being spread across more parameters. We attribute the drop in performance for the non-private LLM with number of maps to optimization difficulties and local minima as well as higher sensitivity to hyperparameters. A small number of maps (between 2 and 5) is sufficient for this datasets, especially in the private setting. Our LLMs attain $82.8 \pm 0.5\%$ (non-private), $82.0 \pm 0.4\%$ ($\epsilon \approx 1.5$), and $79.8 \pm 0.4\%$ ($\epsilon \approx 0.2$) compared to $84 \pm 0.5\%$ for a non-private DNN.

In Fig. 7 we consider an example from each class and show the weighted linear maps by the LLM for each example as well as its integrated gradients (IG) (Sundararajan, Taly, and Yan 2017). For our LLM we consider the non-private and two private cases. In general, there is good agreement between all attribution methods; they are relatively sparse and focus on a small set of features. We found that IG varied much more between examples from the same class than our LLM. For strong privacy ($\epsilon < 0.1$), the linear maps are much less sparse, highlighting the trade-off between interpretability and privacy.

5 Related Work

Interpretability. The *saliency* map and *gradient-based attribution* methods are one of the most popular explanation methods that identify relevant regions and assign importance to each input feature (e.g., pixel for image data) (Selvaraju et al. 2016; Ribeiro, Singh, and Guestrin 2016; Smilkov et al. 2017; Sundararajan, Taly, and Yan 2017; Montavon et al. 2015; Bach et al. 2015; Ancona et al. 2017).
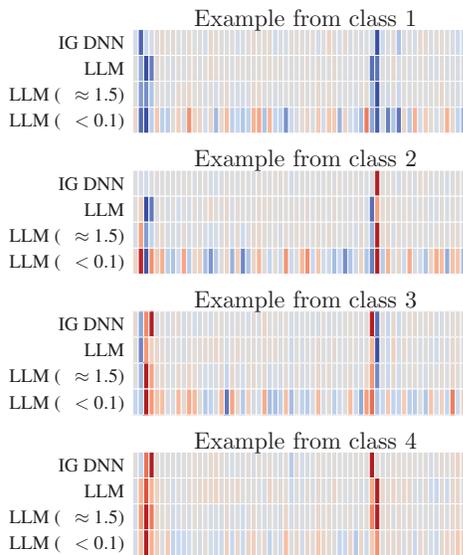
Figure 7: Integrated gradient (IG) and weighted linear filters (LLM; our method) for all 62 feature for one example from each class from the Henan Renmin dataset. For LLM we consider the non-private case (LLM) as well as two private cases with strong ($\epsilon < 0.1$) and weaker ($\epsilon \approx 1.5$) privacy. Entries are normalized and colorcoded between blue $= -1$, gray $= 0$, and red $= 1$.

These methods typically use first-order gradient information of a complex model with respect to inputs, to produce maps that indicate the relative importance of the different input features for the classification. An obvious downside of these approaches is that they provide explanations conditioned on only *a single* input and hence it is necessary to manually assess each input of interest in order to draw a class-wide conclusion. In contrast, our approach can draw class-wide conclusions without manually assessing each input, because it outputs the most relevant explanations in terms of a collection of linear maps for each class. For explanations conditioned on any specific input, our model can provide an input-dependent weighted collection of these features related to that specific input.

Privacy. To privatize complex models, such as deep neural networks, a popular approach is to add noise to the gradients in the stochastic gradient descent (SGD) algorithm (Abadi et al. 2016; Papernot et al. 2017; McMahan et al. 2017). An alternative approach is to directly perturb the objective with additive noise (Zhang et al. 2012; Phan et al. 2016; Phan, Wu, and Dou 2017). In these works, the objective function is approximated by the Taylor expansion, and the resulting coefficients of the polynomials are perturbed before training. We found the latter approach less practical than the former, as we need to choose which order of polynomial degree to use. Typically, adding more layers introduces a more nested-ness in the objective function requiring a higher order polynomial for accurate approximation.

A high degree of polynomial approximation, however, increases the privacy loss as the dimensionality of the coefficients grow. From our perspective, the gradient perturbation method is simple to use and model agnostic. Recently proposed methods for private training through ensembles of teacher models (Papernot et al. 2017; 2018) are less useful to us here, as they consider a special setting where some non-private data is available in addition to the private dataset. Our method distinguishes itself by making interpretability a key component of the trained model and does not rely on access to additional public data.

Lastly, LLMs are reminiscent of *Mixture of experts* (ME) models. MEs assign different specialized linear models to different parts of input space in a discriminative task (see Masoudnia and Ebrahimpour (2014) for an overview of existing ME models). In our case, each local expert model is class specific and contributes to a weighted linear map for that class. The weighting provides an input-dependent *significance* for each linear map, and considering more than one map per class increases flexibility to fit the data better. *Mixtures of factor analyzers* (MFA) are also similar to ME models, but for density estimation of high-dimensional real-valued data (Ghahramani and Hinton 1997)

6 Conclusion and Discussion

We proposed a family of simple models which uses several *locally linear maps* (LLM) per class to provide interpretable features in a privacy-preserving manner while maintaining high classification accuracy. Results on two image benchmark datasets as well as a medical dataset indicate that a reasonable trade-off between classification accuracy, privacy *and* interpretability can indeed be struck and tuned by varying the number of linear maps. Nevertheless, several open questions for future research remain. First, the datasets in this paper are still relatively simple, such that it would be intriguing to see the limits of complexity the LLM model can model with a sufficiently high accuracy. Second, the current model does not interact with a larger and richer counterpart, such as a neural network, due to privacy constraints. It would be interesting to investigate if gaining gradient information of a more flexible model at particularly important input points in a DP way would be possible, in order to combine benefits of both models.

References

Abadi, M.; Chu, A.; Goodfellow, I.; Brendan McMahan, H.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. *ArXiv e-prints*.

Ancona, M.; Ceolini, E.; Öztireli, A. C.; and Gross, M. H. 2017. A unified view of gradient-based attribution methods for deep neural networks. *CoRR* abs/1711.06104.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* 10(7):1–46.

Blocki, J.; Blum, A.; Datta, A.; and Sheffet, O. 2012. The johnson-lindenstrauss transform itself preserves differential privacy. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science* 410–419.

Carlini, N.; Liu, C.; Kos, J.; Erlingsson, Ú.; and Song, D. 2018. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *CoRR* abs/1802.08232.

Dwork, C., and Roth, A. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9:211–407.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, 1322–1333. New York, NY, USA: ACM.

Ghahramani, Z., and Hinton, G. E. 1997. The em algorithm for mixtures of factor analyzers. Technical report, University of Toronto.

Goodman, B., and Flaxman, S. 2016. European Union regulations on algorithmic decision-making and a "right to explanation". *arXiv e-prints* arXiv:1606.08813.

Kenthapadi, K.; Korolova, A.; Mironov, I.; and Mishra, N. 2013. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality* 5(1).

LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database.

Li, R.; Liu, W.; Lin, Y.; Zhao, H.; and Zhang, C. 2017. An ensemble multilabel classification for disease risk prediction. *Journal of healthcare engineering* 2017.

Masoudnia, S., and Ebrahimpour, R. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review* 42(2):275–293.

Maxwell, A.; Li, R.; Yang, B.; Weng, H.; Ou, A.; Hong, H.; Zhou, Z.; Gong, P.; and Zhang, C. 2017. Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC Bioinformatics* 18(14):523.

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2017. Learning differentially private language models without losing accuracy. *CoRR* abs/1710.06963.

Montavon, G.; Bach, S.; Binder, A.; Samek, W.; and Müller, K. 2015. Explaining nonlinear classification decisions with deep taylor decomposition. *CoRR* abs/1512.02479.

Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; and Talwar, K. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Erlingsson, U. 2018. Scalable private learning with PATE. In *International Conference on Learning Representations*.

Phan, N.; Wang, Y.; Wu, X.; and Dou, D. 2016. Differential privacy preservation for deep auto-encoders: An application of human behavior prediction. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 1309–1316. AAAI Press.

Phan, N.; Wu, X.; and Dou, D. 2017. Preserving Differential Privacy in Convolutional Deep Belief Networks. *ArXiv e-prints*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1135–1144. New York, NY, USA: ACM.

Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2016. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR* abs/1610.02391.

Shokri, R., and Shmatikov, V. 2015. Privacy-preserving deep learning. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 909–910.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F. B.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *CoRR* abs/1706.03825.

Song, C.; Ristenpart, T.; and Shmatikov, V. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, 587–601. New York, NY, USA: ACM.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. *CoRR* abs/1703.01365.

Voigt, P., and Bussche, A. v. d. 2017. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 1st edition.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv* abs/1708.07747.

Zhang, J.; Zhang, Z.; Xiao, X.; Yang, Y.; and Winslett, M. 2012. Functional Mechanism: Regression Analysis under Differential Privacy. *ArXiv e-prints*.