Private Learning for High-Dimensional Targets with PATE

Dominik Fay[1, 2] dominikf@kth.se
Jens Sjölund [2] jens.sjolund@elekta.com
Tobias J. Oechtering [1] oech@kth.se

[1]KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
[2]Elekta AB, Box 7593, SE-103 93 Stockholm, Sweden

Abstract

Preventing unintentional leakage of information about the training set has high relevance for medical image segmentation because medical scans contain particularly sensitive information. While differential privacy offers mathematically rigorous protection, the high output dimensionality of segmentation tasks prevents the direct application of state-of-the-art algorithms such as Private Aggregation of Teacher Ensembles (PATE). In order to alleviate this problem, we consider the use of dimensionality reduction to map the prediction target into a lower-dimensional latent space to reduce the required noise level during the aggregation stage. To this end, we assess the suitability of principal component analysis (PCA) theoretically and autoencoders experimentally on a brain tumor dataset.

Introduction

Differential privacy (DP) protects participants against an important class of privacy threats by providing a provable lower bound of the optimal adversary's error rate when trying to detect the presence of any single individual (Kairouz, Oh, and Viswanath 2017). One particular advantage of this concrete guarantee is that it is interpretable at an operational level, in contrast to more abstract information-theoretic measures and best-effort obfuscation schemes . Moreover, it has been argued that DP may assist in compliance with privacy legislation, in particular the General Data Protection Regulation in the European Union (Cummings and Desai 2018).

PATE (Papernot et al. 2017; 2018) is a training algorithm that preserves differential privacy and has several attractive properties for medical applications. First, it can be used with decentralized data. This is relevant because aggregating data from several clinics into a centralized dataset is often not an option. Instead, each institution can train its own model on its respective fraction of the dataset. Second, it is model-agnostic, i.e. it allows the local models to take arbitrary form. In particular, it allows all hospitals to use different learning algorithms, e.g. for the purpose of fine-tuning to differences in measurement protocols or abundance of data. Moreover, model size is not penalized which is beneficial for the training of large neural networks.

On the other hand, PATE does penalize target dimensionality. In the case of image segmentation, for instance, a prediction needs to be made for each pixel, meaning that very few images can be labeled for the student if the cumulative privacy cost is to be kept below a reasonable threshold. In our work, we address this limitation by exploiting the correlation between the labels of pixels within a segmentation map. Just like images, segmentation maps generally show some coherent spatial structure (e.g. the shape of a tumor; background), hence the pixel labels are not independently distributed. By removing this redundancy we can represent the map as a vector in a lower-dimensional space where less noise is needed to achieve the same privacy level.

We consider linear and nonlinear methods separately. In the linear case, we use principal component analysis and give a simple analytical form for the mean-square error in the reconstructions, on which the student is trained subsequently. This allows us to choose the number of latent variables optimally such as to find the best trade off between noise reduction and retention of information. In the nonlinear case, we use an autoencoder on which we perform our experimental evaluation.

Background

Differential Privacy

A randomized algorithm $\mathcal{M} : \mathcal{X}^N \to \mathcal{Y}$ is an algorithm that takes as input N instances from a domain $\mathcal{X}$ and outputs a sample from a probability distribution over $\mathcal{Y}$. Differential privacy requires a randomized algorithm to sample from similar distributions for adjacent inputs. Here, adjacency is defined via the Hamming distance: Two inputs $x = (x_1, \dots, x_N)$ and $x' = (x'_1, \dots, x'_N)$ are said to be adjacent if there is one index $n \in \{1, \dots, N\}$ such that $x_n \neq x'_n$. We denote the set of all adjacent pairs by $\mathrm{Adj}(\mathcal{X}^N)$.

Definition 1 (Differential Privacy (Dwork et al. 2006))**.** *A randomized algorithm $\mathcal{M} : \mathcal{X}^N \to \mathcal{Y}$ is said to be (ϵ, δ)-differentially private if for all $(x, x') \in Adj(\mathcal{X}^N)$ and $Y \subseteq$*

$\mathcal{Y}$,

$$\Pr[\mathcal{M}(x) \in Y] \leq e^{\epsilon} \Pr[\mathcal{M}(x') \in Y] + \delta. \quad (1)$$

Differential privacy is typically guaranteed by estimating the influence any single example can have on the algorithm's output (the sensitivity) and then adding just enough noise to obscure this influence.

Definition 2 (Sensitivity)**.** *The sensitivity of a function* $f : \mathcal{X}^N \to \mathbb{R}^M$ *is defined as*

$$S_2(f) = \sup_{(x,x') \in Adj(\mathcal{X}^N)} ||f(x) - f(x')||_2. \quad (2)$$

PATE PATE (Papernot et al. 2017; 2018) is a decentralized differentially private training algorithm for classification problems that relies on privacy-preserving knowledge transfer. The private training set consists of K disjoint subsets, on each of which a so called teacher model is trained. Subsequently, an additional unlabeled public dataset is labeled by the teacher models and a student model is trained based on the teachers' predictions. The teacher votes are collected in a histogram upon which noise is added (e.g. using the Gaussian mechanism) such as to guarantee the desired level of (ϵ, δ)-differential privacy. Since the student is only shown the noisy histograms, the resulting student model also preserves differential privacy.

Dimensionality Reduction

Principal Component Analysis PCA is an orthonormal change of basis that decorrelates data points. The basis vectors are called the principal components. PCA is used for dimensionality reduction by only keeping the first few components, i.e. those that explain most of the variability within the data set. Computationally, the principal components can be found by an eigendecomposition of the sample covariance matrix.

Formally, if we have a matrix $X \in \mathbb{R}^{N \times d}$ whose rows are the data points $x_1^T, \ldots, x_N^T \in \mathbb{R}^d$ then PCA finds an orthogonal matrix $A \in \mathbb{R}^{d \times d}$ whose rows are the principal components $a_1, \ldots, a_d \in \mathbb{R}^d$. If we let $A_l = (a_1, \ldots, a_l)^T$ then the low-dimensional representation z_n of the n-th point can be computed as $z_n = A_l x_n$. The reverse transformation is computed as $\hat{x}_n = A_l^T z_n$, using the fact that $A^T = A^{-1}$ because A is orthogonal.

Autoencoder Autoencoders are neural networks that have a bottleneck layer and are trained to approximate the identity function. The bottleneck layer is the smallest layer in the network and, in particular, has fewer nodes than the input layer. The layers before the bottleneck layer are called the encoder, the following layers the decoder. Since the autoencoder is trained to output its input, it is forced to learn a lower-dimensional representation of the input in its bottleneck layer that preserves as much information as possible. The subsequent layers are only a function of the representation, not of the original input, which excludes the possibility of skip connections, for instance.

When used for image compression, the autoencoder is typically a convolutional network, where the encoder reduces the spatial resolution with each layer through pooling and the decoder increases the spatial resolution through upsampling. Formally, we represent the encoder by a function $f_{enc} : \mathbb{R}^d \mapsto \mathbb{R}^l$ and the decoder by $f_{dec} : \mathbb{R}^l \mapsto \mathbb{R}^d$.

Linear dimensionality reduction in PATE

In this section, we present our dimensionality reduction scheme in the linear case, which allows for a thorough formal analysis. In particular, we show that due to the well known properties of PCA, the squared error after reconstruction has a simple analytical form and the number of principal components l can be chosen in advance such as to trade off the loss of signal due to compression against the reduction of noise optimally.

In the classification version of PATE, teacher predictions are aggregated by counting the votes for each class and adding noise on top of the vote counts. While in principle possible, it would be challenging to apply this approach directly to segmentation by treating each pixel as a separate classification problem because a large number of queries would need to be answered for every single image. Since the queries are locally highly correlated, this would lead to an unnecessarily high (estimation of the) privacy loss. Instead, we can use PCA to obtain a more compact lower-dimensional representation so that fewer queries are needed. The representations are averaged and Gaussian noise is added. The procedure is detailed in Algorithm 1.

Proposition 1. *Algorithm 1 is* (ϵ, δ)*-differentially private.*

Proof sketch. The algorithm is a repeated application of the Gaussian mechanism to the aggregation function $f_{agg}(x) = \frac{1}{K} \sum_{k=1}^{K} A_l y_{nk}$, which has sensitivity

$$S_2(f_{agg}) = \frac{1}{K} \max_{\hat{y} \in [-1,1]^d} ||A_l \hat{y}||_2 \quad (3)$$

$$\leq \frac{\sqrt{d}}{K} \quad (4)$$

for teacher predictions $y_{nk} \in [0, 1]^d$.

We analyze the cumulative privacy loss over repeated invocations using the composition theorem of Rényi differential privacy (RDP) (Mironov 2017). The free parameter that denotes the order of the Rényi divergence can be optimized analytically. After translating the RDP loss back to the DP loss, we can see that the corresponding standard deviation to stay below the chosen loss is

$$\sigma = \frac{\sqrt{Nd}\left(\sqrt{\log \delta^{-1} + \epsilon} + \sqrt{\log \delta^{-1}}\right)}{\sqrt{2} K \epsilon}. \quad (5)$$

The full proof is given in the appendix. □

Note that the sensitivity we use for setting the noise magnitude is only an upper bound. In principle, the maximization problem in (3) can be solved in polynomial time [1]

[1] If we treat l as constant

Data: K teacher models $t_1, \ldots, t_K$; N unlabeled inputs $x_1, \ldots, x_N$; privacy parameters ϵ, δ; truncated principal component matrix $A_l \in \mathbb{R}^{l \times d}$
Result: Student model
for *n = 1 to N* **do**
  for *k = 1 to K* **do**
    Run the teacher model $y_{nk} = t_k(x_n)$
    Compress the prediction $z_{nk} = A_l y_{nk}$
  end
  Draw $\gamma_n \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\sqrt{Nd}\left(\sqrt{\log \delta^{-1} + \epsilon} + \sqrt{\log \delta^{-1}}\right)}{\sqrt{2}K\epsilon}$
  Aggregate and perturb $\bar{z}_n = \frac{1}{K}\sum_{k=1}^{K} z_{nk} + \gamma_n$
  Recover the segmentation $\hat{y}_n = A_l^T \bar{z}_n$
end
Train the student model on $((x_n, \hat{y}_n))_{n=1..N}$

Algorithm 1: PCA-PATE

by exploiting the structure of A_l (Allemand et al. 2001; Karystinos and Liavas 2010) but we found that the bound (4) was close to optimal for the dataset we considered.

Utility of PCA

The simplicity of the transformations that are used in PCA allows us to characterize the squared error after reconstruction. Let $Y = (y_1, \ldots, y_N)$ be the i.i.d. zero-centered data on which PCA is performed with sample covariance matrix $\hat{\Sigma} = YY^T/(N-1)$. PCA finds the (descendingly ordered) eigenvalues $\lambda_1, \ldots, \lambda_d$ and corresponding eigenvectors $A = (a_1, \ldots, a_d)^T$ of $\hat{\Sigma}$. For an unseen point y (the mean of the teacher predictions) from the same distribution, the expected error due to aggregation and perturbation is

$$\mathcal{L}_{\text{PCA}} = \mathbb{E}[||y - A^T(\pi_l A y + \gamma)||_2^2], \tag{6}$$

where π_l is the projection onto the first l entries and $\gamma \sim \mathcal{N}(0, \sigma^2 I)$. After multiplying from the left with A, we can see that the error decomposes into the removal of information due to PCA and noise:

$$\mathbb{E}[||(I - \pi_l)Ay||_2^2] + \mathbb{E}[||\gamma||_2^2] = \mathbb{E}\left[\sum_{j=l+1}^{d} (a_j y)^2\right] + l\sigma^2.$$

Each term in the sum is the variance of y along the respective principal component, for which λ_j is an unbiased estimator, thus $\mathcal{L}_{\text{PCA}} = \sum_{j=l+1}^{d} \lambda_j + l\sigma^2$. We can minimize this error by choosing the number of principal components l such that, loosely speaking, only those directions with more signal than noise are retained:

$$\arg\min_l \mathcal{L}_{\text{PCA}} = \arg\min_{l \in \{1,\ldots,d\}} \left\{\lambda_l | \lambda_l > \sigma^2\right\}. \tag{7}$$

In (soft) segmentation, y consists of probabilities and independent Gaussian noise would violate the constraint that they should sum to 1. Moreover, we are typically interested in the cross entropy, not the squared error. Therefore, we perform the aggregation on the unnormalized logits (e.g. prior to the softmax layer) instead and normalize afterwards. At this stage, the scores are often expected to be approximately Normally distributed, so Gaussian noise should not destroy utility unnecessarily and the squared error is a meaningful measure.

Nonlinear dimensionality reduction in PATE

While PCA is highly interpretable and well grounded in statistical theory, the target may not always be linearly compressible. Furthermore, the arithmetic mean of the low-dimensional representations also corresponds to a linear operation in the original space, which might not be desired. In the case of images, for instance, it can lead to blurry results. If we use an autoencoder instead, we can address both of these shortcomings and additionally gain more control over the sensitivity, e.g. by choosing the activation function appropriately in the bottleneck layer.

While the commonly used tanh or logistic functions could be used to bound the sensitivity, the bound would be imposed by an l-cube – that is, by the max-norm – which is an inefficient use of space when we only need to bound the l_2-norm. For this reason, we choose an activation $\phi : \mathbb{R}^{l+1} \to \mathcal{B}_l$ which maps to the unit l-ball $\mathcal{B}_l = \{x \in \mathbb{R}^l : ||x||_2 \leq 1\}$. The first input is used to determine the distance from the origin and the remaining inputs are normalized by their l_2-norm. A similar activation function has been described in the context of spherical regression (Liao, Gavves, and Snoek 2019). The distance from the origin is calculated by means of a scaled logistic function (to approximate the standard Normal CDF) raised to the $(1/l)$-th power, the rationale being that in a uniform distribution over the unit l-ball, the distance of a randomly chosen point is distributed according to $U^{1/l}$ where $U \sim$ Uniform[0, 1]. In summary, we have

$$\phi(o_0, \ldots, o_l) = \frac{o_{1:l}}{\left(\exp\left(-o_0\sqrt{8/\pi}\right) + 1\right)^{1/l} \sqrt{\sum_{i=1}^{l} o_i^2}}.$$

The hope is then that the autoencoder will learn an approximation to the mapping into this bounded region that preserves the most information.

Algorithm 2 describes our procedure formally. In particular, we use a 3D convolutional autoencoder for the task of brain tumor segmentation. The encoder part consists of 3x3x3 convolutional layers with ReLU activations, followed by max-pooling. The bottleneck layer is a fully-connected layer with activations as described above. By adding Gaussian noise in this layer even during training, the decoder observes the same noisy distribution that it will later perform its predictions on, even though this would not be necessary from a privacy perspective. The decoder part consists of convolutions with ReLU activations, followed by transposed convolutions. The output layer uses softmax activations. Cross-entropy is used as the loss function.

Proposition 2. *Algorithm 2 is (ϵ, δ)-differentially private.*

Proof sketch. By construction, the bottleneck activations have l_2 norm at most 1, which leads to a sensitivity bound of $S_2(f_{agg}) \leq 2/K$. The remainder of the proof is analogous to Proposition 1. □

Data: K teacher models $t_1, \ldots, t_K$; N unlabeled inputs $x_1, \ldots, x_N$; privacy parameters ϵ, δ; encoder f_{enc} and decoder f_{dec}
Result: Student model
for *n = 1 to N* **do**
  for *k = 1 to K* **do**
    Run the teacher model $y_{nk} = t_k(x_n)$
    Compress the prediction $z_{nk} = f_{enc}(y_{nk})$
  end
  Draw $\gamma_n \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\sqrt{2N}\left(\sqrt{\log \delta^{-1} + \epsilon} + \sqrt{\log \delta^{-1}}\right)}{K\epsilon}$
  Aggregate and perturb $\bar{z}_n = \frac{1}{K}\sum_{k=1}^{K} z_{nk} + \gamma_n$
  Recover the segmentation $\hat{y}_n = f_{dec}(\bar{z}_n)$
end
Train the student model on $((x_n, \hat{y}_n))_{n=1..N}$

Algorithm 2: Autoencoded PATE

Experiments

Dataset We evaluate Autoencoded PATE on the brain tumor dataset that is part of the Medical Segmentation Decathlon (Simpson et al. 2019). It consists of 750 volumes acquired by magnetic resonance imaging (MRI), 484 of which are labeled. The data comes from 18 different institutes and was measured using various clinical protocols. All scans are registered on the same grid with a spatial resolution of 1 mm^3. We use 400 of these as training data, of which 320 are split evenly across the $K = 8$ teachers, and 80 are left for the student (without labels). We use the labels of this partition to train the autoencoder. The remaining 84 volumes form the test set. We measure the Dice coefficient (Dice 1945) individually for each class and report the average over the three classes.

Base models For all teachers and the student model, we use a vanilla 3D U-Net with four layers of pooling, similar to the nnU-Net (Isensee et al. 2018) which has been shown to work consistently across a range of medical segmentation tasks. We train on $64 \times 64 \times 64$ patches, on half of which we perform rotations or mirroring for data augmentation.

Baseline We compare our approach against DP-SGD (Abadi et al. 2016) applied to the same U-Net as described above. We clip gradients for each layer at 4 and apply noise with variance $\sigma^2 = 6$. We include 400 volumes in the training set.

Results With the setup as described above, Autoencoded PATE achieves a Dice score of 0.561 under a privacy budget of $(8, 10^{-3})$. On average, the teachers (before aggregation and perturbation) achieve a Dice score of 0.547. That is, the student can use the teachers as an ensemble to improve upon their individual performances, even though noise is added. This observation is in accordance with the original work on PATE (Papernot et al. 2017; 2018) on classification datasets. With DP-SGD, we measure a Dice score of 0.53 under the same privacy budget. While we have not fine-tuned all hyperparameters and architecture choices (neither for PATE nor for DP-SGD), the relative performance difference is still meaningful because the same network has been used in all methods.

Conclusion

In this work, we have explored the use of dimensionality reduction to answer high-dimensional queries in the context of PATE. In the case of PCA, the error can be described analytically and the number of principal components can be chosen optimally in terms of the mean squared error. For autoencoders, we have presented a suitable architecture and activation function for the bottleneck layer that can use the space that is bounded by the l_2 norm efficiently. Experimentally, we have described initial work applying Autoencoded PATE to brain tumor segmentation. By using simple convolutional networks, we ensured comparability between methods. However, we expect that higher performance and/or a lower privacy bound can be achieved with more complex architectures, which we intend to address in future work.

Acknowledgements

This paper is partially based on initial work that has been done in (Fay 2019).

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Allemand, K.; Fukuda, K.; Liebling, T. M.; and Steiner, E. 2001. A polynomial case of unconstrained zero-one quadratic optimization. *Mathematical Programming* 91(1):49–52.

Atchison, J., and Shen, S. M. 1980. Logistic-normal distributions: Some properties and uses. *Biometrika* 67(2):261–272.

Cummings, R., and Desai, D. 2018. The role of differential privacy in GDPR compliance. *FATREC '18*.

Dice, L. R. 1945. Measures of the amount of ecologic association between species. *Ecology* 26(3):297–302.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Fay, D. 2019. Membership privacy in neural networks for medical image segmentation. Master's thesis, KTH Royal Institute of Technology.

Isensee, F.; Petersen, J.; Klein, A.; Zimmerer, D.; Jaeger, P. F.; Kohl, S.; Wasserthal, J.; Koehler, G.; Norajitra, T.; Wirkert, S.; et al. 2018. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*.

Kairouz, P.; Oh, S.; and Viswanath, P. 2017. The composition theorem for differential privacy. *IEEE Transactions on Information Theory* 63(6):4037–4049.

Karystinos, G. N., and Liavas, A. P. 2010. Efficient computation of the binary vector that maximizes a rank-deficient quadratic form. *IEEE Transactions on information theory* 56(7):3581–3593.

Liao, S.; Gavves, E.; and Snoek, C. G. 2019. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9759–9767.

Mironov, I. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 263–275.

Muller, M. E. 1959. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM* 2(4):19–20.

Papernot, N.; Abadi, M.; Úlfar Erlingsson; Goodfellow, I.; and Talwar, K. 2017. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations (ICLR)*.

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

Simpson, A. L.; Antonelli, M.; Bakas, S.; Bilello, M.; Farahani, K.; van Ginneken, B.; Kopp-Schneider, A.; Landman, B. A.; Litjens, G.; Menze, B.; et al. 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*.

Appendix

Privacy proof

First, we recall the basics of Rényi differential privacy (RDP) that we will make use of, which have been shown in (Mironov 2017).

Proposition 3 (Gaussian mechanism)**.** *Let $f : \mathcal{X}^N \to \mathbb{R}^M$ be a function with sensitivity $S_2(f)$ and $\mathcal{N}(\mu, \Sigma)$ the multivariate Normal distribution. If $\gamma \sim \mathcal{N}(0, \sigma^2 I)$ then the mechanism $\mathcal{M}(d) = f(d) + \gamma$ satisfies $(\alpha, \frac{\alpha S_2(f)^2}{2\sigma^2})$-RDP for any $\alpha > 1$.*

Proposition 4 (Composition)**.** *If a randomized mechanism $\mathcal{M}_1$ satisfies (α, ϵ_1)-RDP and $\mathcal{M}_2$ satisfies (α, ϵ_2)-RDP then their composition $(\mathcal{M}_1, \mathcal{M}_2)$ satisifies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

Proposition 5 (From RDP to DP)**.** *If a randomized mechanism $\mathcal{M}$ satisfies (α, ϵ)-RDP then it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP for any $0 < \delta < 1$.*

In the case of PCA, the sensitivity of the aggregation is

$$S_2(f_{mean}) = \max_{y, y' \in [0,1]^{K \times d}} \left|\left| \frac{1}{K} A_l \sum_{k=1}^{K} (y_k - y'_k) \right|\right| \quad (8)$$

Since y and y' are allowed to differ in only one index k, we can equivalently write

$$S_2(f_{mean}) = \frac{1}{K} \max_{\hat{y} \in [-1,1]^d} ||A_l \hat{y}|| \,. \quad (9)$$

As A_l is constructed by removing rows from an orthogonal matrix, it acts as a rotation and/or reflection followed by $(d - l)$ perpendicular projections onto a coordinate axis. None of these operations can increase the norm of the vector that is being transformed, hence

$$S_2(f_{mean}) \leq \frac{\sqrt{d}}{K}. \quad (10)$$

Therefore, for any $\alpha > 1, \sigma > 0$, $\mathcal{M}$ satisfies $(\alpha, \frac{\alpha d}{2\sigma^2 K^2})$-RDP by Prop. 3, which scales to $(\alpha, \frac{\alpha N d}{2\sigma^2 K^2})$-RDP after N invocations by Prop. 4, which implies (ϵ, δ)-DP with

$$\epsilon = \frac{\alpha N d}{2\sigma^2 K^2} + \frac{\log 1/\delta}{\alpha - 1} \quad (11)$$

by Prop. 5.

For a fixed δ, we can choose α in (11) such as to minimize ϵ. Defining auxiliary variables $a = \frac{Nd}{2\sigma^2 K^2}, b = \log \delta^{-1}$ and setting the derivative to zero

$$\frac{d\epsilon}{d\alpha} = a - b(\alpha - 1)^{-2} \stackrel{!}{=} 0$$

yields

$$\alpha = \sqrt{\frac{b}{a}} + 1. \quad (12)$$

Clearly, $\alpha > 1$ and is thus a valid minimizer. It is globally optimal due to the convexity of ϵ with respect to α. Resubstituting α into (11):

$$\epsilon = a + 2\sqrt{ab} \quad (13)$$

$$= \frac{Nd}{2\sigma^2 K^2} + \sqrt{2 \log \delta^{-1} \frac{Nd}{\sigma^2 K^2}}. \quad (14)$$

Finally, we need to know which noise level σ is required to satisfy DP for given ϵ, δ. To answer this, we solve (14) for σ by multiplying both sides with σ^2. The resulting quadratic equation is solved by

$$\sigma = \frac{\sqrt{Nd}\left(\sqrt{\log \delta^{-1} + \epsilon} + \sqrt{\log \delta^{-1}}\right)}{\sqrt{2} K \epsilon}, \quad (15)$$

which is the noise level we choose in Algorithm 1.

Activation function

The activation function

$$\phi(o_0, \ldots, o_l) = \frac{o_{1:l}}{\left(\exp\left(-o_0 \sqrt{8/\pi}\right) + 1\right)^{1/l} \sqrt{\sum_{i=1}^{l} o_i^2}}$$

is designed to give an approximately uniform distribution over $\mathcal{B}_l$ whenever the inputs are standard Gaussians. The uniform distribution is desirable because it has the highest information content (entropy) for distributions over bounded

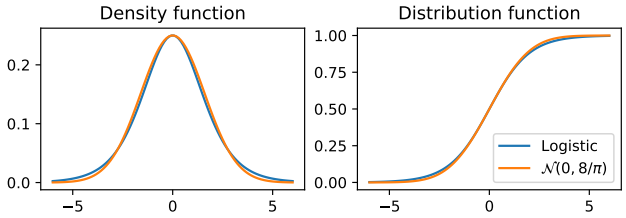

Figure 1: Comparison of Logistic and Normal distribution

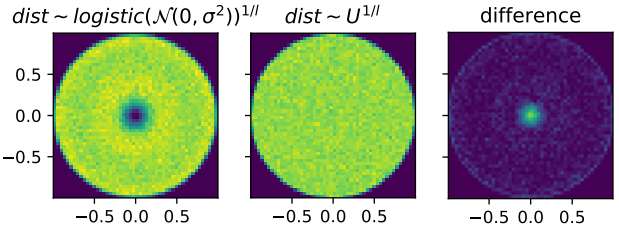

Figure 2: Samples from the distribution of activations ($l = 2$) for standard Normal inputs (left) compared to a true uniform distribution (center) and their absolute difference (right).

support. The normalization factor $\sqrt{\sum_i o_i^2}$ maps $o_{1:l}$ uniformly onto the unit $(l-1)$-sphere (Muller 1959) while $\exp\left(-o_0\sqrt{8/\pi}\right)+1$ maps o_0 to the Logit-Normal distribution (Atchison and Shen 1980). For $\sigma^2 = 8/\pi$, the Normal distribution approximates the standard Logistic distribution well (see Figure 1), thus the corresponding Logit-Normal is approximately uniform over $[0, 1]$. Figure 2 shows the distribution of activations for standard Normal inputs for the two-dimensional case.