

Certiably Robust Interpretation via Differential Privacy

Presented by Pratiksha Kamble(201673777)

Primary Supervisor: Wiley Ruan

Secondary Supervisor: Xiaowei Huang





Introduction



- Certifiably robust interpretation is a critical aspect of ensuring the reliability and trustworthiness of machine learning models' explanations.
- Differential privacy is a privacy-preserving mechanism that safeguards individual data while enabling accurate analysis.
- We'll explore how differential privacy enhances interpretation robustness and the significance of this approach in real-world applications.

Aims and Objectives

AIMS

- To train and test Neural Networks for Differential Privacy
- To Develop interpretation technique using Differential Privacy

OBJECTIVES

Develop a Robust Smooth-based interpretation technique

Prove that Renyi-Differential Privacy has a better performance

Achieve a balanced relationship between accuracy and computational efficiency.

Differential Privacy



What Is Differential Privacy?

- **Privacy Shield:** Keeps individual information hidden in data studies.
- **Adding "Noise":** Mixes up data a bit so no one's info stands out.
- **Secret Keeper:** Makes sure results don't reveal personal details.

Why We Need Differential Privacy?

- **Data Safety:** Stops private info from being stolen or misused.
- **Better Than Hiding Names:** More secure than just removing personal details.
- **Following Rules:** Keeps up with laws that protect personal information.
- **Building Trust:** Makes people feel safe to share their data.

Renyi-Differential Privacy



Rényi Differential Privacy is implemented by using the `DPGradientDescentGaussianOptimizer` from TensorFlow Privacy, which applies a Gaussian noise multiplier to gradients and clips the gradients' L2 norm as part of the optimization process during training to protect the data privacy.

```
noise_multiplier = 1.1
num_microbatches = batch_size # Setting num_microbatches to the same size as batch_size for simplicity
learning_rate = 0.001
epsilon = 0.5 # Desired epsilon value for privacy guarantee
delta = 1e-5 # Desired delta value for privacy guarantee

optimizer = DPGradientDescentGaussianOptimizer(
    l2_norm_clip=l2_norm_clip,
    noise_multiplier=noise_multiplier,
    num_microbatches=num_microbatches,
    learning_rate=learning_rate
)

loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True, reduction=tf.keras.losses.Reduction.NONE)

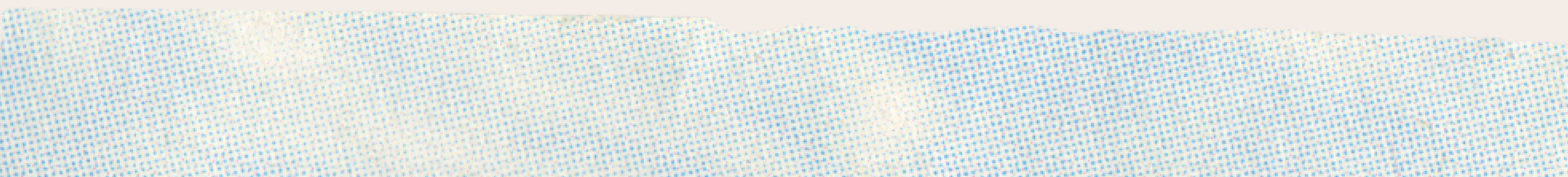
@tf.function
def train_step(images, labels):
    logits = model(images, training=True)
    per_example_loss = loss_fn(labels, logits)
```

Trials:

1. Privacy-First Model Trials:

- Testing VGG16 and CNNs with Differential Privacy safeguards.
- Focused on secure, private data use in model training.

2. Efficiency vs. Protection:

- Examining the balance between strong privacy and fast computation.
 - Seeking optimal trade-offs for swift and private data processing.
- 

Challenges with Differential Privacy:

1. Complex Parameter Tuning:

- Finding the right balance for parameters such as noise multiplier, privacy budget (ϵ), and delta (δ) can be non-trivial.

2. Model Performance:

- Adding noise to achieve differential privacy often degrades model **accuracy**, especially for complex models.

3. Computational Overhead:

- Differential privacy mechanisms, including RDP, can introduce significant computational overhead, increasing training time.
- Optimizing the trade-off between privacy and computational resources can be difficult, especially for large-scale applications.

```
Epoch 50/50  
step 0: loss = 2.3029  
step 100: loss = 2.3020  
Accuracy over epoch 50: 0.0954
```


Optimization of Renyi-Differential Privacy

- **Dataset Utilized:** CIFAR10, normalized to [0, 1]
- **Model Architecture:** Customized VGG16, adapted for CIFAR10 input shape (32x32x3)
- **Privacy Method:** Differential Privacy via **DPGradientDescentGaussianOptimizer**
- **Differential Privacy Parameters:**
 - L2 norm clipping: **1.0**
 - Noise multiplier: **1.1**
 - Number of microbatches: equal to **batch_size**
- **Learning Rate Schedule:**
 - Initial learning rate: **0.5**
 - Decay steps: **5000**
 - Decay rate: **0.9**
 - Implemented via custom **ExponentialDecay**
- **Privacy Budget:** Epsilon (ϵ): **0.35**, Delta (δ): **1e-5**
- **Batch Processing:**
 - Batch size: **250**

Case: Certifiably Robust Interpretation

Original	Original	Original
		
Perturbed	Perturbed	Perturbed
		

Image 1 Cosine Similarity between original and perturbed top-2 attributions: [-0.99999994 -1.0000001]
Image 2 Cosine Similarity between original and perturbed top-2 attributions: [-1. -1.]
Image 3 Cosine Similarity between original and perturbed top-2 attributions: [-1.0000001 -0.99999994]

Evaluation

Evaluation Factor	Description	Measurement Method
Model Accuracy and privacy	Assess the balance between model accuracy and privacy guarantees.	Training accuracy metric; privacy budget (ϵ , δ).
Learning Rate Optimization	Examine the custom scheduler's impact on model training.	Convergence rate; learning rate at each epoch.
Computational Efficiency	Record the computational cost of training with differential privacy.	Time per epoch; GPU/CPU/memory usage
Stability Against Perturbations	Evaluate model robustness to input alterations.	Cosine similarity of predictions before and after perturbation
Image Attribution Consistency	Analyse the model's consistency in identifying key image features	Consistency of top-k attributions for original vs. perturbed images.
Visual Inspection	Qualitatively compare original and adversarial perturbed images.	Visual comparison via displayed image pairs.

✖ Changes to Original Proposal



1. Change of Dataset to avoid complexity:

- **Simplicity of Images:** CIFAR-10 consists of 60,000 32x32 color images which are simpler and smaller in size compared to the high-resolution images in Pascal VOC.
- **Computational Efficiency:** Due to smaller image sizes and the less complex nature of CIFAR-10, models trained on it can be iterated more rapidly, facilitating the tuning of differential privacy parameters which often requires extensive experimentation.

REFERENCES

- 1.. Liu, A. et al. (2022) 'Certifiably robust interpretation via Rényi differential privacy', Artificial Intelligence, 313, p. 103787. doi:10.1016/j.artint.2022.103787
- 2.
- 3.Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., & Bolukbasi, T., 2021. Guided Integrated Gradients: an Adaptive Path Method for Removing Noise. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5048-5056. <https://doi.org/10.1109/CVPR46437.2021.00501>.
- 4.Shrikumar, A., Su, J., & Kundaje, A., 2018. Computationally Efficient Measures of Internal Neuron Importance. ArXiv, abs/1807.09946.
- 5.Khan, M., Kwon, S., Choo, J., Hong, S., Kang, S., Park, I., Kim, S., & Hong, S., 2020. Automatic detection of tympanic membrane and middle ear infection from oto-endoscopic images via convolutional neural networks. Neural networks: the official journal of the International Neural Network Society, 126, pp. 384-394. <https://doi.org/10.1016/j.neunet.2020.03.023>.
- 6.Zhang, Y., Hong, D., McClement, D., Oladosu, O., Pridham, G., & Slaney, G., 2021. Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. Journal of Neuroscience Methods, 353. <https://doi.org/10.1016/j.jneumeth.2021.109098>.



Thank You
