Homework Assignment 4

Response by: **Prathvish Mithare (7028692)**

Due: **2:00pm Tuesday, 8 February 2024 on CISPA CMS**

Collaboration Policy: You should do this assignment by yourself and submit your own answers. You may discuss the problems with anyone you want and it is also fine to get help from anyone on problems with LaTeX or Jupyter/Python. You should note in the *Collaborators* box below the people you collaborated with.

Collaborators: Monseej Purkayastha (7047530), Sohum Rajesh Davale (7024001)

Implementation Problems. Below are two implementation problems that you need to provide your code implementations and results in Jupyter notebook. For the tasks which require explanations or descriptions (e.g., Task 3 of Problem 2), you can provide your answers in the submitted PDF.

Problem 1 (15 pts) In this problem, we will discuss and implement the *SmoothGrad* method for producing saliency map visualizations. The method is first introduced in the paper "SmoothGrad: removing noise by adding noise". Specifically, consider an input $\boldsymbol{x} \in \mathbb{R}^d$ and a classifier $f_\theta : \mathbb{R}^d \to \mathcal{Y}$ that we aim to probe, where $\mathcal{Y}$ denotes the set of class labels. Let $F_\theta : \mathbb{R}^d \to \mathbb{R}$ be function mapping from the input layer to the logit with respect to the predicted class $\hat{y} = f_\theta(\boldsymbol{x})$. *SoomthGrad* computes the saliency maps of F_θ at $\boldsymbol{x}$ based on the following rule:

$$\text{SmoothGrad}(F_\theta, \boldsymbol{x}; m, \sigma) = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\boldsymbol{x}+\boldsymbol{\delta}_i} F_\theta(\boldsymbol{x} + \boldsymbol{\delta}_i), \text{ where } \boldsymbol{\delta}_i \sim \mathcal{N}(0, \sigma^2).$$

Here, m represents the number of Gaussian samples and $\sigma > 0$ denotes the standard deviation parameter.

Task 1 (5pt): Answer the following question: How will the two parameters m and σ affect the computed saliency maps? Please also provide a brief explanation about your answer.

Solution

SmoothGrad aims to estimate the local average within a high-dimensional input space for computing saliency maps. Due to the computational complexity of directly computing this average, it relies on generating multiple samples (typically denoted as m) to approximate it effectively. The parameter m governs the precision of the saliency maps: increasing m enhances accuracy, while decreasing it leads to less accurate maps.

Moreover, to smooth out the gradient and compute the local average, each sample is perturbed by adding noise sampled from a normal distribution $N(0, \sigma^2)$, where σ is a parameter controlling the size of the smoothing kernel. Adjusting σ influences the extent of smoothing applied to the saliency maps. While increasing σ can enhance the accuracy of the smoothed maps, excessive smoothing beyond a certain threshold may render the maps ineffective due to over-smoothing.

Task 2 (10pt): Now we implement the *SoomthGrad* method on a neural network and CIFAR-10 images. We will use the *pretrained ResNet18 model* as the classifier f_θ, and you need to randomly sample 5 testing CIFAR-10 images. You can refer to Problem 2 of Homework Assignment 3 on how to prepare the CIFAR-10 dataset. In particular, the number of Gaussian samples should be set as $m = 50$, and the the standard deviation parameter $\sigma \in \{0, 0.05, 0.1, 0.2, 0.3, 0.5\} * (x_{\max} - x_{\min})$, where $x_{\max}$ and $x_{\min}$ represent the largest and the smallest pixel value of the correpsonding sampled CIFAR-10 image $\boldsymbol{x}$.

Visualize the saliency maps as heatmaps for each combination of the 5 sampled CIFAR-10 images and the 6 σ parameters, and also visualize the sampled CIFAR-10 images. The produced figure should be in the similar format to Figure 3 of the "SmoothGrad" paper. In addition, briefly discuss the results (i.e., whether they are aligned with your answers for Task 1).

Solution

From Figure 1, we can see that adding noise does improve the legibility of the saliency maps. We see that injecting too much noise results in illegible maps which are useless for any kind of interpretation.

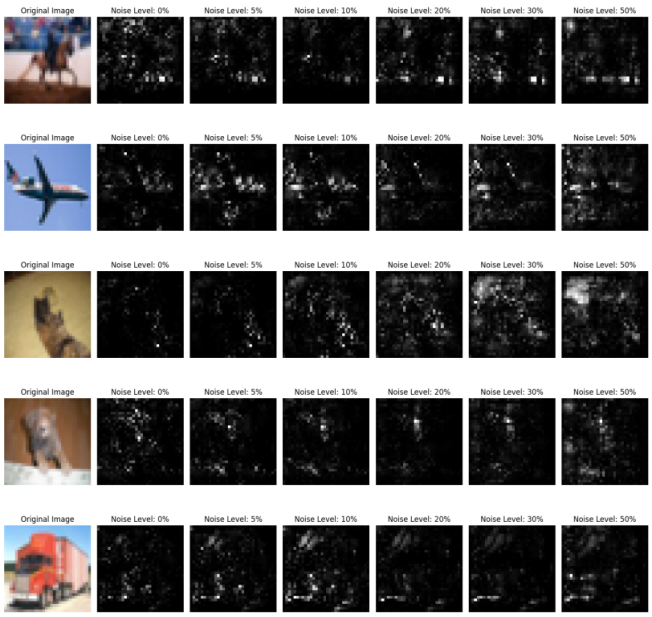

Figure 1: SmoothGrad on Cifar-10

Problem 2 (25 pts) In this problem, we will consider a simple setting of out-of-distribution robustness on the MNIST dataset. First of all, a ood_mnist.ipynb file is provided in the jupyter_notebook folder. It provides the sample code for generating customized MNIST digits: *"0"* and *"8"*. The code is adapted from the Github repository: ec23-tutorial created by AIRI, where you can find the source code of all the functions. First, you should understand the content provided in the ood_mnist.ipynb file.

Instead of considering two MNIST digits and random forest, we will extend the results to the multi-class classification task of classifying ten MNIST digits using neural network. Specifically, you need to set the neural network using the CNN architecture with 4 convolution layers and 2 MLP layers, the same architecture considered in Problem 3 of Homework Assignment 2. You can set the optimizer of SGD with learning rate 0.1, and train the model for 30 epochs.

Task 1 (5 pts): Construct three additional MNIST testing datasets: *colored_MNIST*, *grayscaled_MNIST*, and *rotated_MNIST*. To be more specific, *colored_MNIST* consist of colored testing MNIST image by randomly applying one of the three color transformations (i.e., red, green and blue) provided in the function gen_colored(), *greyscaled_MNIST* consist of grayscaled MNIST testing images by randomly applying one of the three transformations provided in function gen_weighted_grayscale(), while *rotated_MNIST* consists of rotated MNIST testing images by applying the transformation using the function rr(). All the functions are detailed in the ec23-tutorial repo. All the testing datasets should contain 10000 images.

Task 2 (5 pts): Train a neural network with the required architecture and learning scheduler on the original MNIST training dataset. Report the mean accuracy and standard deviation over 3 repeated trials on each of the three constructed MNIST testing datasets and the original MNIST testing dataset. Do you observe an obvious accuracy drop when tested on customized datasets? Also, please provide a brief discussion about the results.

	Original	Coloured	Grayscale	Rotation
Mean Accuracy	99.02%	97.73%	9.93%	44.9%
Standard Deviation	0.000497	0.007061	0.002734	0.002748

Figure 2: Mean Accuracy and Standard Deviation of the model over 3 trials

Solution

According to the data presented in Figure 2, there is a significant decrease in accuracy observed for the grayscaled and rotated datasets. In the case of colored images, it is hypothesized that color variations do not significantly impact the image beyond minor disturbances that may not be visually discernible. This could explain the slight decline in accuracy observed for the colored dataset.

Task 3 (5 pts): Next, we want to improve the out-of-distribution robustness performance of our learning algorithm. One typical method for OOD robustness is to apply various types of data augmentations to the training images. Read the paper "AUGMIX: A Simple Data Processing Method to Improve Robustness and Uncertainty", then summarize the existing data augmentation techniques and their proposed AUGMIX method that is designed for enhancing the model robustness against distribution shift.

Solution

Current augmentation strategies focus on specific aspects such as addressing robustness issues arising from data shifts. However, synthetic augmentations often struggle to generalize to unforeseen shifts occurring in real-world scenarios, impacting model calibration. To mitigate these challenges, a widely adopted approach is data augmentation, involving the application of various augmentation techniques to training data to enhance model robustness. Traditional augmentation methods encompass actions like left-right flipping and cropping, while newer approaches include techniques like Cutout (removing patches from images), CutMix (replacing removed patches with patches from other images), Mixup (blending images elementwise), and Gaussian Patch (injecting noise into random image regions).

AugMix introduces a novel data augmentation approach that enhances both model robustness and uncertainty estimation, seamlessly integrating into existing training workflows. By applying multiple simple augmentation operations and layering them together, AugMix generates a single robust example with diverse transformations, facilitating the training of models with varied inputs. In contrast to previous methods that directly chain different augmentation primitives to improve diversity, AugMix's layering approach prevents images from quickly degrading or straying off the data manifold. Additionally, the authors advocate for the adoption of the Jensen-Shannon Divergence Consistency Loss, ensuring model stability and consistency across a broad spectrum of inputs.

Task 4 (10 pts): Adapt the data augmentation techniques discussed in the paper to improve the out-of-distribution robustness on MNIST. Instead of considering our constructed synthetic datasets, we consider a better benchmark dataset created for evaluating OOD robustness for MNIST, named MNIST-C, which is introduced in the paper "MNIST-C: A Robustness Benchmark for Computer Vision". The dataset consists of a variety of corrupted MNIST images, which can be downloaded here: https://github.com/google-research/mnist-c. To simplify the task, you only need to consider three types of corruptions (out of the total 15 types), i.e., *Motion Blur*, *Fog*, and *Zigzag*. Describe how you augment the training data for improving OOD robustness. Report the testing accuracy of the vanilla model trained in Task 2 and the model learned with your proposed augmentation technique on each of the three corrupted datasets.

Solution

	Motion Blur	Fog	Zig-zag
Clean model	94.2%	69.51%	90.08%
Augmented model	95.92%	77.26%	89.26%

Figure 3: Accuracy on MNIST-C dataset for clean and augmented models

Problem 3 (bonus, 5 pts) Read the paper "Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization" and write a one-page short review about the paper. You can follow the general structure of the paper reviews you are supposed to submit on Feb 1st.

Summary

The paper "Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization" empirically demonstrates a strong correlation between in-distribution and out-of-distribution performance across various models and distribution shifts. This correlation holds across different model architectures, hyperparameters, training set sizes, and training durations. The paper also explores cases where this correlation is weaker and provides a theoretical explanation based on a Gaussian data model. This work significantly contributes to understanding machine learning performance in unseen, out-of-distribution environments.

Review

The paper addresses the problem of understanding the correlation between in-distribution and out-of-distribution generalization in machine learning models. This is a significant issue as machine learning models are often deployed in real-world scenarios where they encounter data that is different from the distribution they were trained on.

Previous works primarily focused on in-distribution generalization and did not extensively study the correlation with out-of-distribution performance. This paper empirically demonstrates a strong correlation across various models, distribution shifts, and hyperparameters. It also provides a theoretical explanation based on a Gaussian data model, which was not done in prior works.

The strengths of the paper lie in its extensive empirical analysis and the breadth of scenarios it covers. It also provides a theoretical explanation, which is a significant contribution. However, the paper's weakness might be that it does not fully explain why the correlation is weaker in some cases. Also, the Gaussian data model used for the theoretical explanation might not be applicable to all types of data.

The theoretical explanation based on the Gaussian data model might be difficult to understand for readers without a strong background in statistics or machine learning theory. The paper could have provided more intuitive explanations or visualizations to help understand this part.

The paper could delve deeper into the cases where the correlation is weaker and provide more insights into these scenarios. It could also explore other theoretical models beyond the Gaussian data model.

The findings of this paper have significant implications for the deployment of machine learning models in real-world scenarios, and further research could explore how to leverage this correlation to improve out-of-distribution performance.

End of Homework Assignment 4 (PDF part)
Don't forget to submit your nice Jupyter notebook!