Understanding Models Through The Training Data

Rebecca Adaimi and Ronshee Chawla

March 31, 2019

1 Introduction

An important question that arises in some various machine learning applications is: "How did the model come to this prediction?" or "How was my training data helpful or harmful for a given prediction?". Koh et al. [2] tried to address these questions by using influence functions to study the model and understand its prediction through its training data. They presented four tasks in which this tool can be used. For our minor project, we focused on the following tasks: (1) understanding model behavior and (2) debugging domain mismatch. We replicated the author's work for each task and extended their analysis by experimenting using other datasets and other models. Sections 2 and 3 present the first and second task respectively. Section 4 presents some of the observations and challenges we faced, and section 5 concludes our report.

2 Understanding Model Behavior

A use case of influence functions is understanding model behavior and how a model reaches a certain prediction. In other words, what were the training points that were most influential (responsible) for this prediction?

2.1 Replication

In the paper, the authors demonstrated this task by comparing two models: (a) inception v3 network with the top layer frozen and (b) an SVM with an RBF kernel on dog and fish images extracted from ImageNet. The models are trained to classify dogs vs. fish. Using the replication code provided, we were able to run this experiment and get the same results as reported in the paper (Figure 1).

2.2 Additional Experiments

In order to observe more the way influence functions can be used to understand model behavior, we implemented some additional experiments.

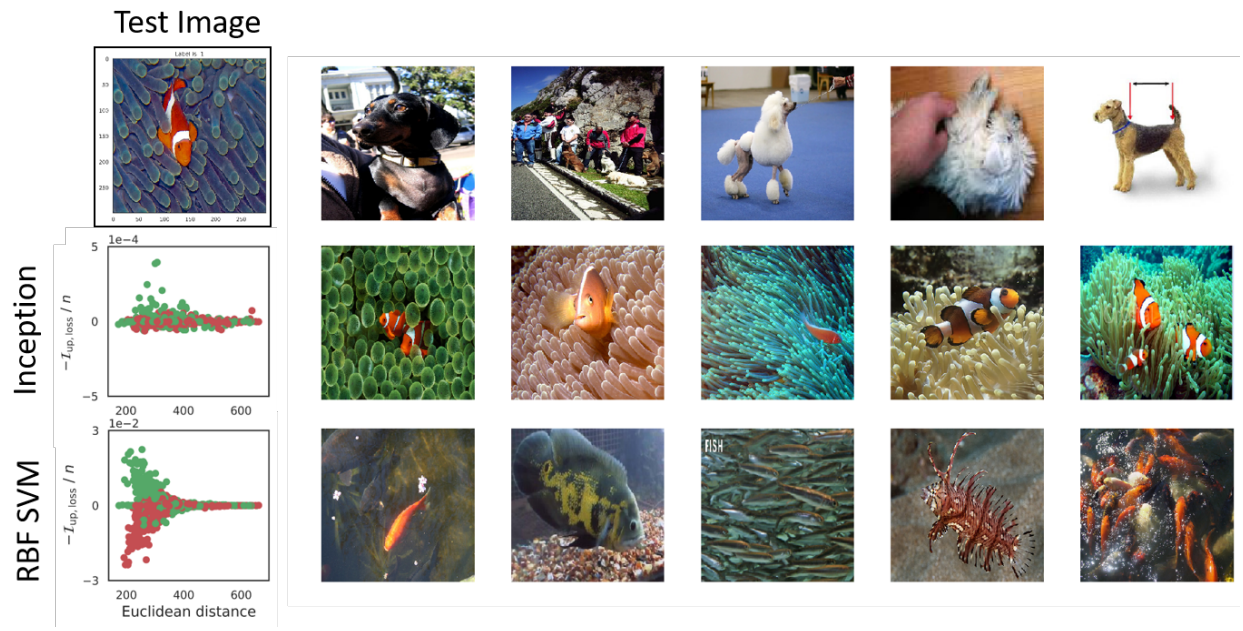

Figure 1: Inception vs. RBF SVM. **Bottom left**: $-\mathcal{I}_{up,loss}(z, z_{test})$ vs. euclidean distance. Green dots are fish and red dots are dogs. **Bottom right**: The 5 most helpful training images, for each model, on the test. **Top right**: 5 images of dogs in the training set that helped the Inception model correctly classify the test image as a fish.

2.2.1 Inception ResNet V2

Using the same dog-fish dataset, we decided to observe the model behavior of an additional model, the Inception ResNet V2 [5]. Inception ResNet V2 is a variation of the Inception V3 network with added residual connections. In some cases, this model outperformed the Inception V3 network. Thus, we wanted to compare the behavior of each model for a certain prediction. To that end, we implemented the Inception ResNet V2 following a similar structure the authors used for the Inception V3 network. Testing on the same test image of a fish, we observe the top images in the training data that influenced the prediction (Figure 2). We can see how each model was influence by some common training images and also different ones. We also did the same analysis but for a test image of a dog (Figure 3). We observed similar behavior across the different models. Inception ResNet as well as Inception were able to capture the distinctive characteristics of the dog in the test image, whereas RBF SVM superficially matched patterns in the training images.

2.2.2 MNIST Dataset

An additional experiment we implemented included classifying handwritten digits, specifically the digits 1 vs. 7, from the MNIST dataset. To that end, we used a binary logistic regression model to classify the handwritten digits and observed the top training samples that were responsible for the prediction of the given test image (Figure 4). Looking at the figure, there does not seem to be any correlation between the influences and distance. Moreover, both digits 1 and 7 could be helpful or harmful for correctly classifying the test image

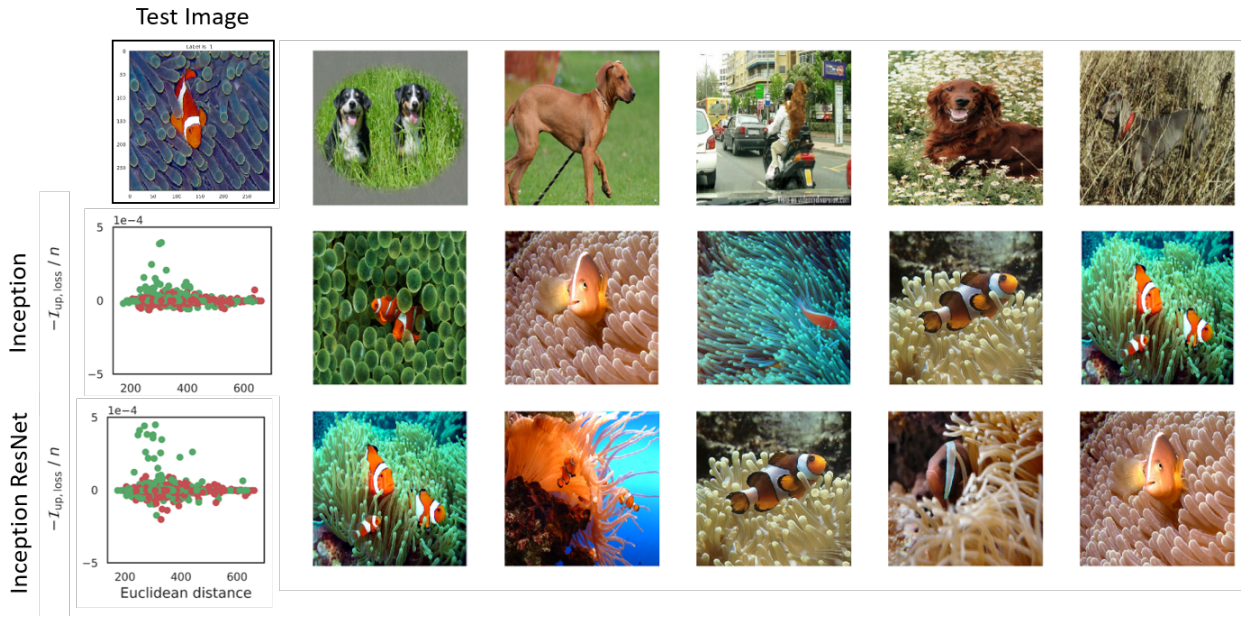

Figure 2: Inception vs. Inception ResNet V2. **Bottom left**: $-\mathcal{I}_{up,loss}(z, z_{test})$ vs. euclidean distance. Green dots are fish and red dots are dogs. **Bottom right**: The 5 most helpful training images, for each model, on the test. **Top right**: 5 images of dogs in the training set that helped the Inception ResNet V2 model correctly classify the test image as a fish.

as a 7.

3 Debugging Domain Mismatch

Domain mismatch occurs when the training distribution does not faithfully represent the overall population, thus resulting in poor accuracy for the test data. In this section, we will show how influence functions can be used to identify the training examples most responsible for errors, which can help identify domain mismatch.

3.1 Replication

In the paper, the authors demonstrated this application by using logistic regression to predict whether a child under age 10 will be readmitted to hospital on a balanced dataset of 20k diabetic patients from 100+ US hospitals [4]. Using the code provided for replication, we were able to replicate this experiment and obtained the same results as reported in the paper.

3.2 Additional Experiments

Following the procedure used by the authors for identifying domain mismatch, we used the same approach on two more datasets: UCI Adult Dataset [1] and UCI Bank Marketing Dataset [3]. The details of the experiments on these datasets are summarized below.

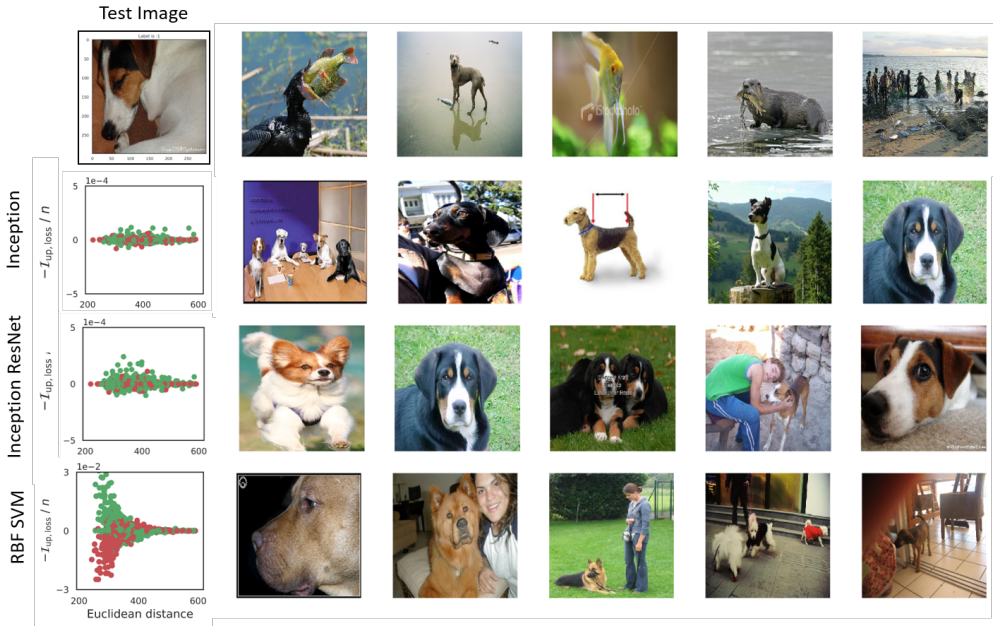

Figure 3: RBF SVM vs. Inception vs. Inception ResNet V2. **Bottom left**: $-\mathcal{I}_{up,loss}(z, z_{test})$ vs. euclidean distance. Green dots are dogs and red dots are fish. **Bottom right**: The 5 most helpful training images, for each model, on the test. **Top right**: 5 images of dogs in the training set that helped the Inception ResNet V2 model correctly classify the test image as a dog.

3.2.1 UCI Adult Dataset

In the UCI Adult Dataset, we introduced domain mismatch by removing 1094 black people having annual income $\leq$ \$50k from the training data, leaving only one black person with annual income $\leq$ \$50k. Using logistic regression, the aim is to predict whether a person has an annual income of $\leq$ \$50k or otherwise. The baseline test to look at the learned features didn't reveal anything: the black indicator variable was not even among the top 20 coefficient values of learned parameters. We then picked a random black person from the test set that was incorrectly classified and calculated $-\mathcal{I}_{\text{up,loss}}(z_i, z_{test})$ for each training instance z_i. We identified the top 10 training examples (by magnitude) which contributed most to the positive and negative influence. It turned out that the only black person with annual income $\leq$ \$50k in the training set had the highest positive influence, whereas training samples of black people with annual income $>$ \$50k had very negative influences. Furthermore, the black indicator variable contributed significantly to the magnitude of $\mathcal{I}_{\text{up,loss}}$ as evident from the calculation of $\mathcal{I}_{\text{pert,loss}}$ on the examples contributing most to the influence.

3.2.2 UCI Bank Marketing Dataset

In the UCI Bank Marketing Dataset, we introduced domain mismatch by removing 5229 people having a housing loan who did not accept the bank term deposit from the training data, leaving only one person having a housing loan who did not accept the bank term deposit. Using logistic regression, the aim is to predict whether a person will accept the

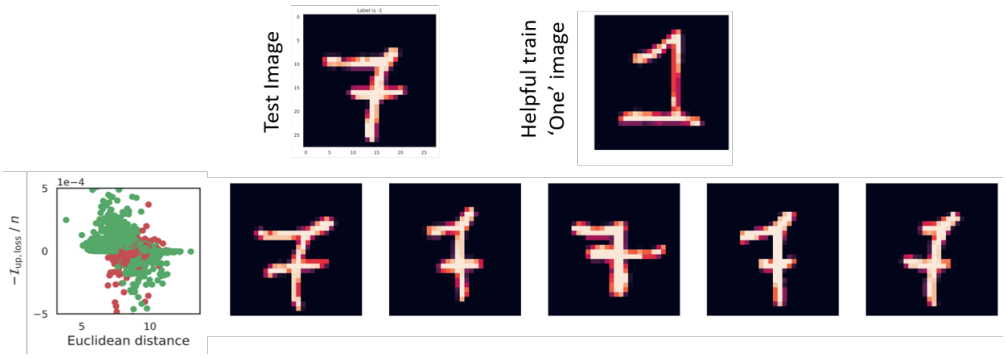

Figure 4: Binary Logistic Regression. **Bottom left**: $-\mathcal{I}_{up,loss}(z, z_{test})$ vs. euclidean distance. Green dots are the digit 7 and red dots are the digit 1. **Bottom right**: The 5 most helpful training images on the test. **Top right**: An image of the 1 digit in the training set that helped the model correctly classify the test image as a 7.

bank term deposit or not. The baseline test to look at the learned features reveals that the indicator variable for having a housing loan had the highest coefficient value among learned parameters. We then picked a random person having a housing loan from the test set that was incorrectly classified and calculated $-\mathcal{I}_{\text{up,loss}}(z_i, z_{test})$ for each training instance z_i. We identified the top 10 training examples (by magnitude) which contributed most to the positive and negative influence. It turned out that the only person having a housing loan who did not accept the bank term deposit in the training set had the highest positive influence, whereas training samples of people who did not have housing loan and accepted the bank term deposit had very negative influences. Furthermore, the indicator variable for having a housing loan contributed significantly to the magnitude of $\mathcal{I}_{\text{up,loss}}$ as evident from the calculation of $\mathcal{I}_{\text{pert,loss}}$ on the examples contributing most to the influence.

4 Challenges and Observations

In the domain mismatch task applied to the hospital readmission dataset, it is easier to see that the training examples which contributed the most in influencing the test loss were those of 4 children. In case of UCI Adult Dataset and UCI Bank Marketing Dataset, similar observations were made depending on the experiments performed on them. However, the difference is not very clear from the difference in test loss for those top 10 training samples (after inducing domain mismatch), because there are thousands of samples corresponding to black people in UCI Adult Dataset and people having a housing loan who accepted the bank term deposit in UCI Bank Marketing Dataset. In order to understand it in a better way, we identified top 100 training examples contributing to negative influence (instead of top 10) and it turns out that most of those training examples were of the same type as the top 10 examples. Moreover, for the UCI Adult Dataset, the Hessian for the loss function was not positive definite for some of the test examples which were classified incorrectly after inducing domain mismatch. Therefore, the results obtained for those test examples were not in accordance with the intuition as explained above. In general, for both UCI Adult Dataset and UCI Bank Marketing Dataset, there were more examples of one class compared to other,

therefore we didn't have a balanced training dataset. This resulted in all the test samples belonging to just one class.

5 Conclusion

We utilized the concept of influence functions in the context of understanding model behavior and debugging domain mismatch. These aforementiond tasks were replicated as given in [2] and also tried on other datasets. We mentioned key observations and challenges encountered while performing these tasks. The code and the data for replicating our experiments is available at add link to code: either create github repo or share code on drive

References

[1] D. Dua and C. Graff. UCI machine learning repository. University of California Irvine, School of Information and Computer Science, 2019.

[2] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1885–1894. JMLR.org, 2017.

[3] S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems, pages 22–31. Elsevier, 2014.

[4] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. In *BioMed Research International*, 2014, 2014.

[5] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2016.