Label Noise

Final project – Machine Learning

course lecture Dr Liad Gottlib

By: Shira Baron and Hadar Baron

Abstract

“Data scientists spend 80% of their time cleaning and manipulating data and only 20% of their time actually analyzing it.”

Collecting large training datasets, annotated with high-quality labels, is costly and time-consuming. Indeed, labels may be polluted by label noise, due to; insufficient information, expert mistakes and encoding errors. The problem is that errors in training labels that are not properly handled may deteriorate the accuracy of subsequent predictions, among other effects. Many works have been devoted to label noise and this paper will provides a concise and comprehensive introduction to this research topic. In particular, it reviews the types of label noise, their consequences and the difference between how algorithms deals with label noise.

Introduction

In classification, it is both expensive and difficult to obtain reliable labels, yet traditional classifiers assume and expect a perfectly labelled training set. This paper reviews the classification problem with popular algorithms of machine learning when data is unclean and has labeling noise.

Mislabelling may come from different sources.

First, the available information may be insufficient to perform reliable labelling language is too limited or if data are of poor quality. Second, even experts

often make mistakes during labelling. Third, classification is in some cases subjective, which results in inter-expert variability. For example, the pattern boundaries provided by two experts for the segmentation of electrocardiogram signals are often slightly different. In addition, incorrect labels may come from communication or encoding problems; real-word databases

are estimated to contain around five percent of encoding errors.

Three types of noise are distinguished here.

First, label noise completely at random (NCAR) occurs independently of the true class and of the values of the instance features. Second, label noise that occurs at random (NAR) depends only on the true label. This can be used to model situations where some classes are more likely to be mislabelled than others. Third, label noise not at random (NNAR) is the more general case, where the mislabelling probability also depends on the feature values. This allows us

to model labelling errors near the classification boundaries.

This paper focus on two types of noise, NCAR – also called ‘symmetric noise’,

And NNAR -called ‘asymmetric noise’.

It reviews a multiclass dataset named CIFAR-10.

Methods

This article examines how basic and popular machine learning, algorithms deal with labeling problems.

The algorithms selected for this paper were ADA-BOOST, RANDOM FOREST, LINEAR REGRESSION and SVM (in this algorithm we did not get results due to runtime issues). These algorithms were learned and expanded in class, these are classic and familiar algorithms in machine learning.

In addition in the article there is an extension of the topic to a niche in machine learning called deep learning. Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input.

dealing with labeling issues and how The article examines a small CNN's deep learning copes with respect to known algorithms in machine learning.

Experiments

In this project we wanted to examine how the algorithms deal with labeling problems. The data set we will be dealing with is called CIFAR-10. Multi-class data consisting of 60,000,000 images of 10 types of class, for each class 10,000 images. When the train is allotted 50000 pictures and the test 10000. We assume that this data set has no labeling issues (although it is almost certain that like all data there are anomalies). The code was written and made in Google Colab, in the language of Python, with the help of libraries known for using the various algorithms. CNN was built ourselves with the help of open source and our add-ons to suit the problem.

First, the algorithms will run without disrupting the data. Then, the data will be disrupted in a controlled manner for a certain percentage of the train. The problem is divided into two types symmetrical noise and asymmetrical noise, when in each one will make a disruption on a certain percentage of the train and then will run the algorithms. The data is disrupted five times in each of the noise types.
10%, 20%, 30%, 40%, 50%

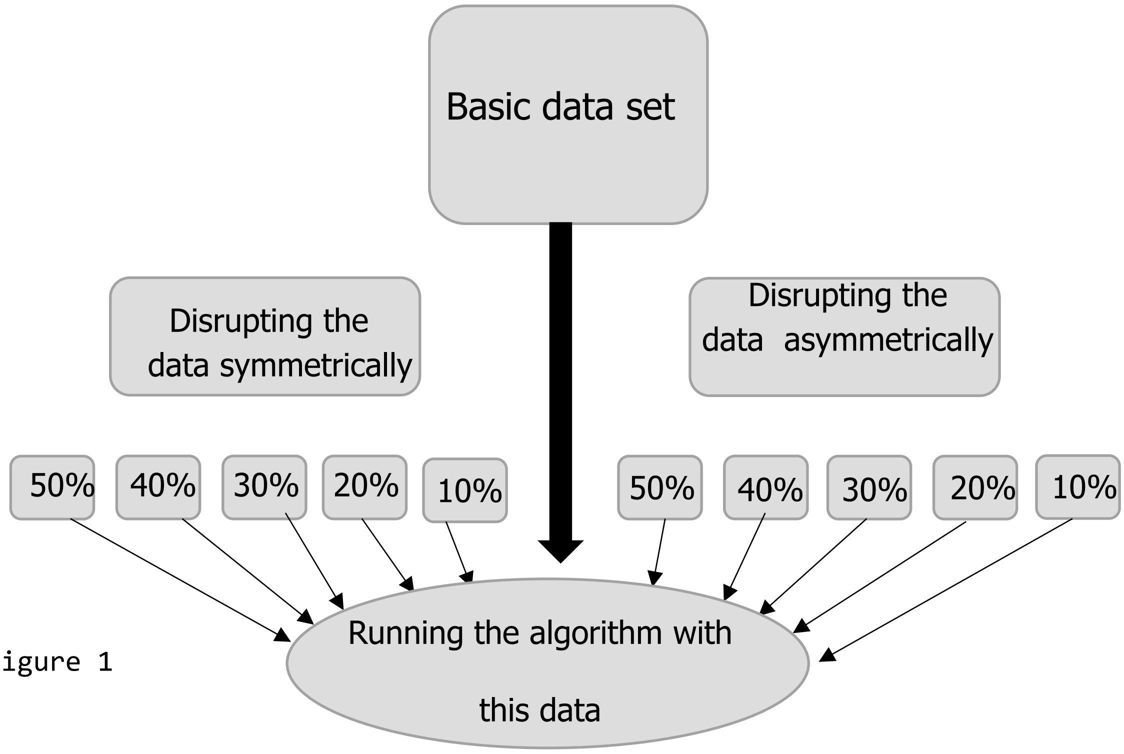
Basic data set
Disrupting the data symmetrically
Disrupting the data asymmetrically
50%
40%
30%
20%
10%
50%
40%
30%
20%
10%
Running the algorithm with this data

Figure 1

This diagram simplifies the process. First the unchanged data is transmitted directly
to the algorithm. In addition, the data is disrupted symmetrically and asymmetrically on a certain percentage of the train data, and the algorithms are being run on each of these percentages.

The data was disrupted manually, with the help of a code symmetrically.
Each time we damaged a certain percentage of the train tag.
symmetrically we went through one of the images and randomly tagged it with another image, 1->6 1->8 3->5 5->2 5->7 9->3…
And asymmetrically we systematically brought it to a controlled error, when one class would be mistakenly labeled as another specific class and so on.
2<->3 5<->6

The results of the algorithms

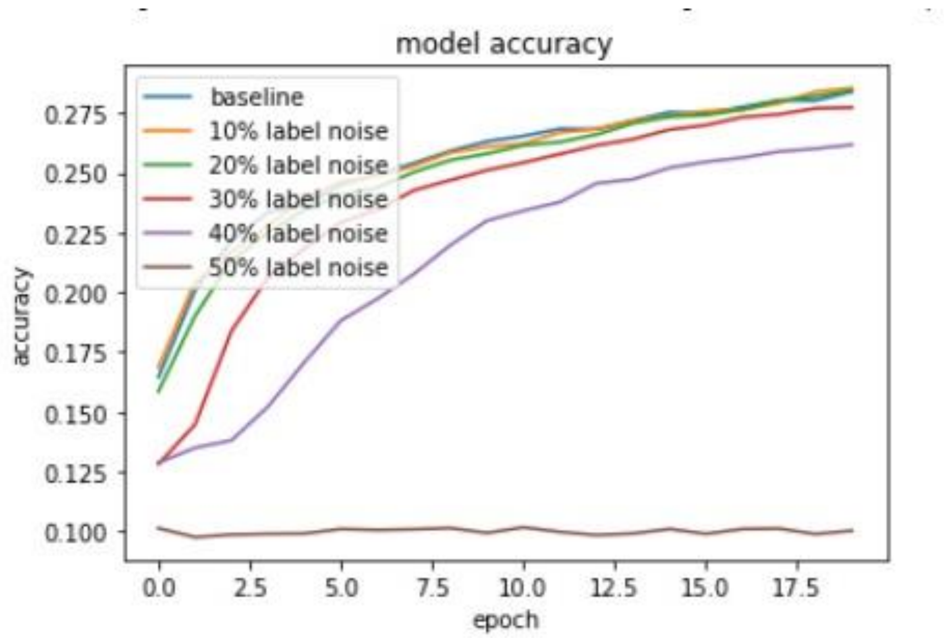

Figure 2 (CNN symmetric)

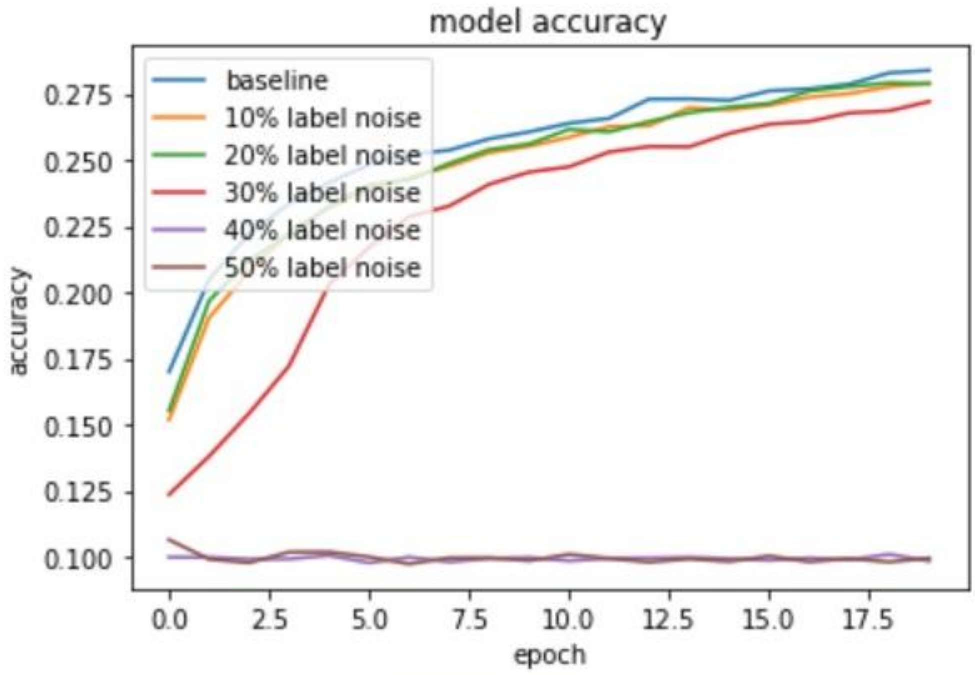

Figure 3 (CNN asymmetric)

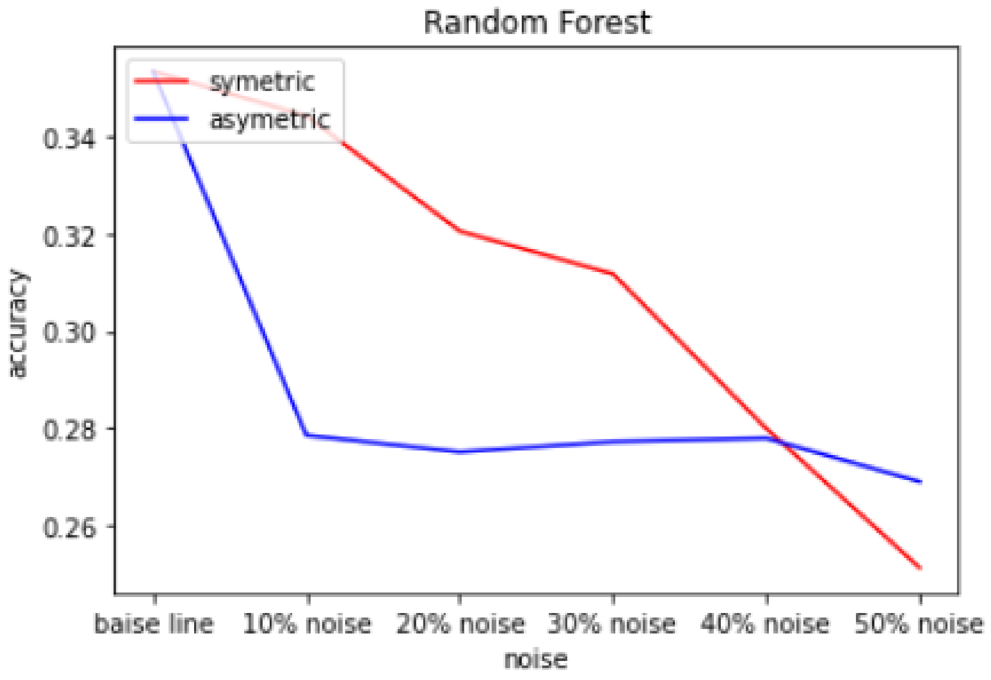

Figure 4 (Random Forest)

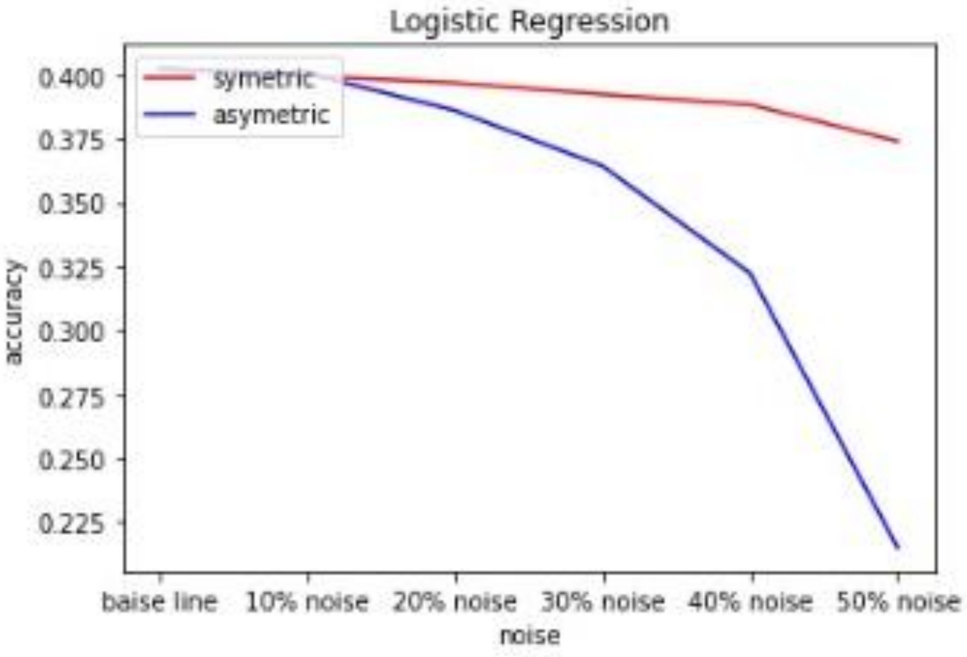

Figure 5 (logistic regression)

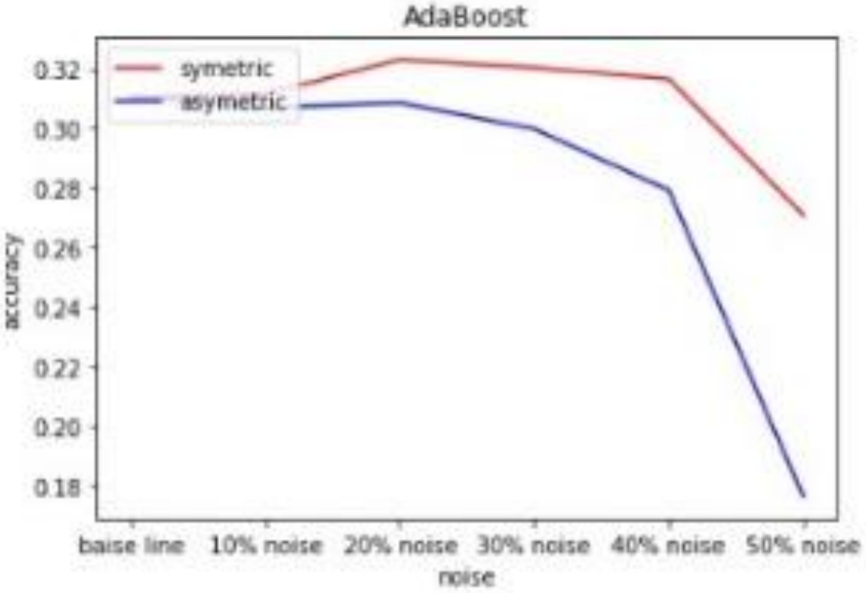

Figure 6 (Ada boost)

Discussion and Analysis

First of all you can see in all the graphs the level of accuracy is not high and that is because we are dealing with the problem of multiclass.

From the analysis of `Figure 2` and `Figure 3` we can clearly see the differences between symmetrical and asymmetrical noise. The network collapses earlier with asymmetric noise, this can be due to the fact that the network process causes the creation of a particular pattern. Because the noise is consistent a wrong pattern is formed and therefore in this test it falls drastically and asymmetrically.

From a general analysis of `Figure 4`, `Figure 5`, and `Figure 6`, it can be seen that the one who managed to deal with the problem best in symmetrical noise was the Logistic Regression algorithm. It can be seen that even in 50% noise there isn't a big effect on the level of accuracy. But with asymmetric noise, the algorithm responded in the least good way and the level of accuracy dropped relatively drastically to the other algorithms. The logistic regression algorithm joined by the Ada boost also failed to stabilize the level of accuracy, unlike the Random forest, which managed to reach a higher level of accuracy by fifty percent than the rest. Other articles show that although random forest brought relatively good results from the rest, the algorithm does not know how to deal with labeling noise. There are some improvements to the algorithm and articles are explaining the changes made to the algorithm and how it helps the algorithm not to be affected by noise.

One might think that the network is "disappointing" in its results relative to the simple algorithms of machine learning. Today networks are the best known and most useful tool in the industry. Despite this, the network brought less good results than the other algorithms. There are many reasons for this, one of which is that the network built for this article may have been too weak and if a strong network had been built the network would have given much better results than the rest. Besides, the problem can be considered simple and the network is a solution to complicated problems.

Conclusion

In this research project, we have expanded and enriched the knowledge in everything related to the basics of machine learning. We understood in depth why and how important it is for the data to be tagged correctly, we were exposed to a major problem in the industry, an unsolvable problem. The topic made us go deep into the algorithms we learned. Also, we were exposed to a new topic that we did not learn in class and that is deep learning, familiarity with concepts, writing code, and creating a network. That has given us exposure to an interesting and contemporary world.

references

1. https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2014-10.pdf
2. https://openaccess.thecvf.com/content_CVPR_2019/html/Li_Learning_to_Learn_From_Noisy_Labeled_Data_CVPR_2019_paper.html
3. https://www.digitalvidya.com/blog/data-cleaning-techniques/#:~:text=Data%20cleansing%20or%20data%20cleaning,the%20dirty%20or%20crude%20data.
4. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
5. https://ieeexplore.ieee.org/abstract/document/554195
6. https://arxiv.org/abs/2003.10471