Label Noise

Oren Zauda

July 18, 2021

Abstract

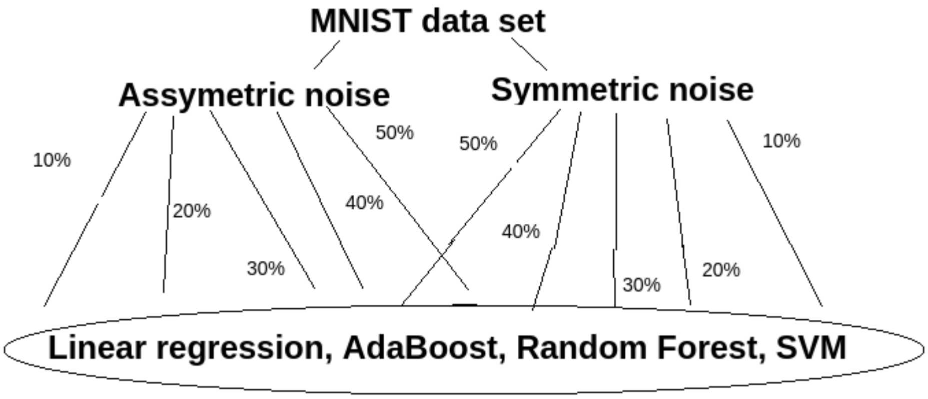
Data scientists spend 80% of their time cleaning data rather than creating insights. Collecting large training data sets, annotated with high-quality labels, is costly and time-consuming. Indeed, labels may be polluted by label noise, due to; insufficient information, expert mistakes and encoding errors. 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.

1 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 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. 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 Mnist, which is a classic dataset of handwritten digits.

2 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. 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.



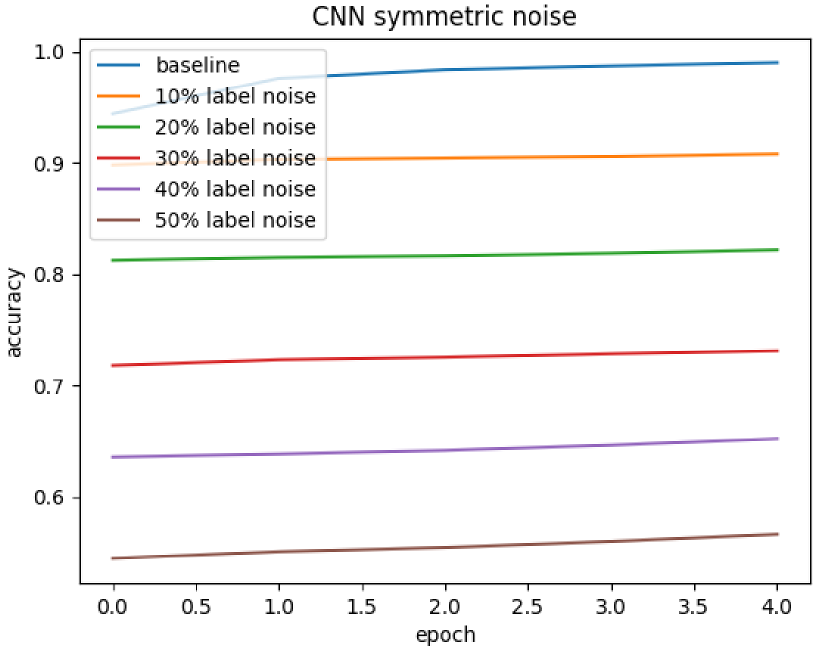
3 Experiments

In this project I wanted to examine how the algorithms deal with labeling problems. The data set I will be dealing with is called MNIST. The MNIST database contains 60,000 training images and 10,000 testing images, both with of 32X32X3 dimension.

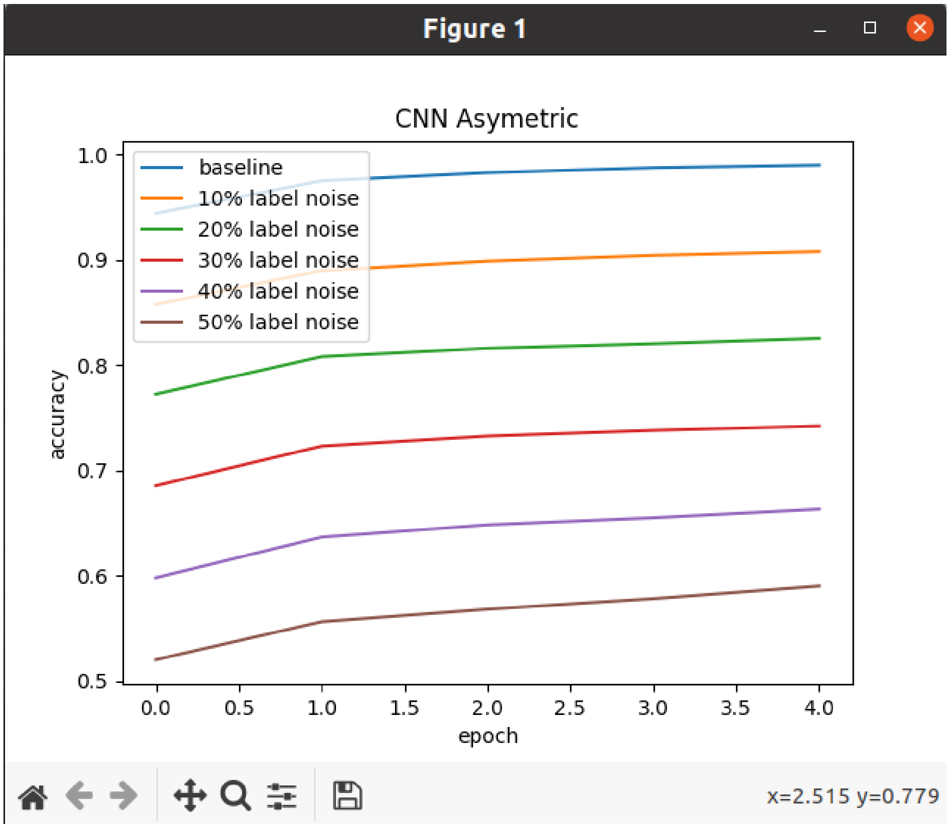
I assume that this data set has no labeling issues (although it is almost certain that like all data there are anomalies). The code was written and made in Visual Studio Code, in the language of Python, with the help of libraries known for using the various algorithms. CNN was built by me with the help of open source and some additions to suit the problem. First, the algorithms will run without disrupting the data. Then, the data will be polluted 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 pollution on a certain percentage of the labels in the train set, and then will run the algorithms. The data is disrupted five times in each of the noise types. The percentage of wrong labels are 10%, 20%, 30%, 40%, 50%.

The symmetric noise obtained by randomly changing the labels to something else. The asymmetric noise obtained by changing some fraction of the labels to another labels consistently, for example, label 3 allways change to label 4, label 9 allways changed to label 2 etc.

4 Results



CNN symmetric noise
baseline
10% label noise
20% label noise
30% label noise
40% label noise
50% label noise
accuracy
1.0
0.9
0.8
0.7
0.6
epoch
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0



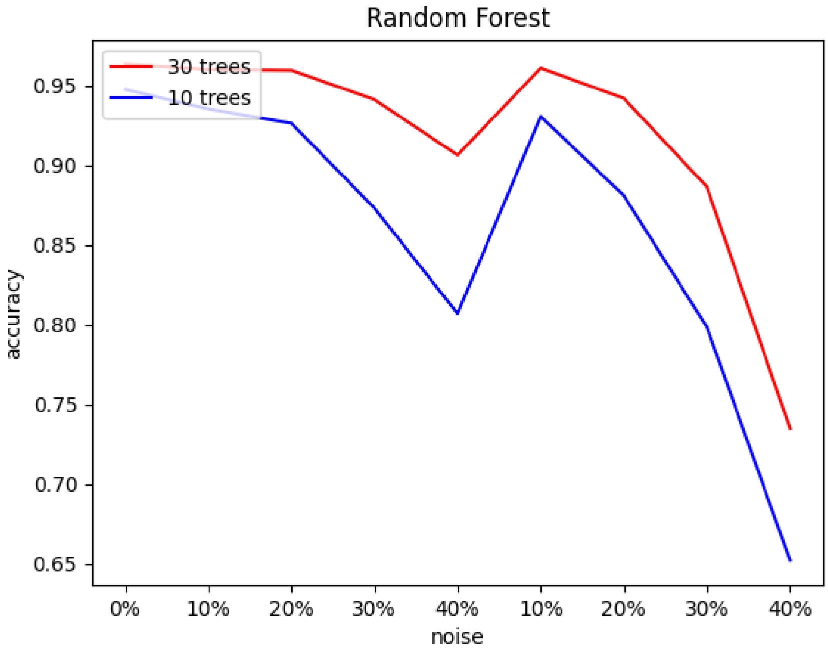
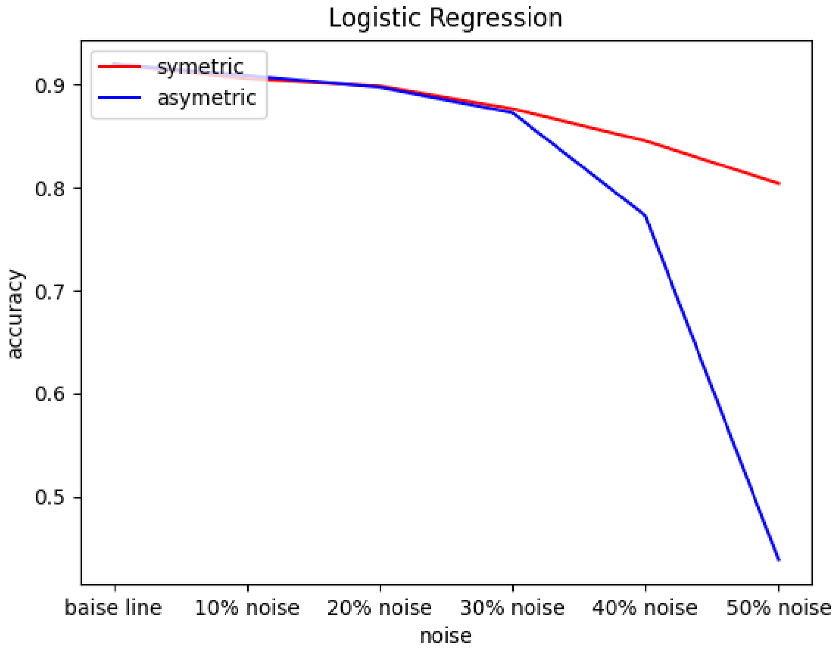
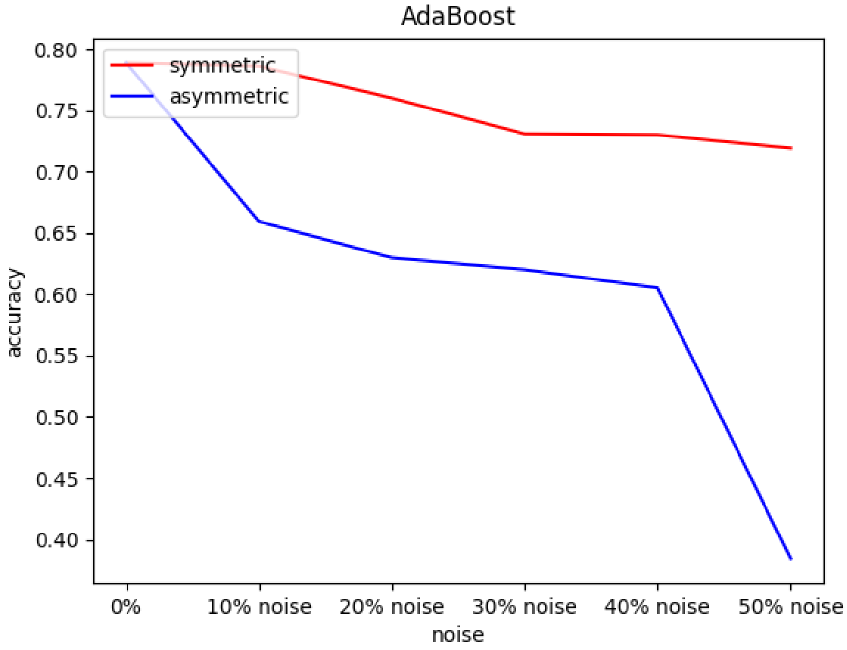
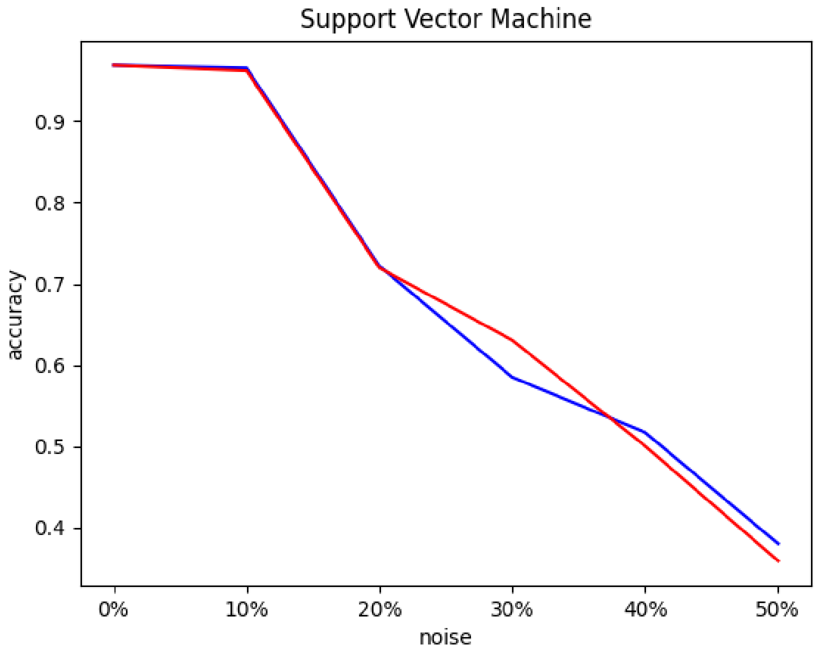


Figure 1: The first 0-40 % relate to a symmetric noise, and the other 0-40% relate to asymmetric noise


Logistic Regression
symetric
asymetric
0.9
0.8
0.7
0.6
0.5
accuracy
baise line
10% noise
20% noise
30% noise
40% noise
50% noise
noise


AdaBoost
symmetric
asymmetric
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
accuracy
0%
10% noise
20% noise
30% noise
40% noise
50% noise
noise


Support Vector Machine
accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0%
10%
20%
30%
40%
50%
noise

5 Discussion and Analysis

- It is clear from all the figures that all the algorithms, except SVM and CNN, cope better with symmetric noise rather than asymmetric noise. The reason for that is the asymmetric noise is consistent, and thus models will learn more from it.

- The best performance with symmetric noise made by logistic regression. The reason for that is the mechanism of the algorithm, it is uses "one vs. all" method. In this method we predicting the probability for a picture to be in any class, and then we pick the class with the highest probability to be our prediction. Label noise can harm this model only if the noise lead the model to predict that another class has higher probability than the real class, which is very unlikely, since in symmetric noise the noise spread equally between the samples and should not bring a specific class to be with higher probability of the real class. In asymmetric noise, the scenario of increasing the probability of another class to be higher than the real class is more likely, because the noise consistently change the real label to another specific label. With a big amount of noise (in my results it is 40%), another class will get the highest prediction, and the accuracy will drop.

- Random forest is probably the best option for any kind noise. The results here show that a forest with only 30 threes perform on training set with 30% symmetric noise and 20% asymmetric noise almost at 90% accuracy. The main idea of this algorithm is to take the majority of the results from the decision threes. Therefore we can "drown" the noise as much as we can increase the number of threes. Here we train pretty big training set with 60,000 pictures, and we used only 30 threes, and we managed to take awful labeling with 20% or 30% noise to a good 90% accuracy. The bad news are that high number of threes is causing too much computing resources. In my computer i could not train more than 300 threes.

- Support vector machine algorithm performs similarly when the noise was symmetric and asymmetric. This is happened probably because the algorithm procedure, it is just find the line or an hyperplane that separates the data with the largest margin. Therefore the consistency of the noise (which is practically the difference between symmetric and asymmetric noise) is not a factor.

- Support vector machine performs with 10% noise almost the same as without noise. I can assume the reason for that, it is probably because of the soft margin. it tolerates a few dots to get misclassified. The noise probably "hides" as a misclassified dots.

- Just like the logistic regression, the performance of support vector machine are very poor when the noise is 40% and 50%. When half or so of the dots in the graph of the training set are off, there is no Feasibility to find a line or an hyperplane that separates the data correctly and predict the class of unseen data in the right way.

6 Conclusion

In this research project, I was expanded and enriched the knowledge in everything related to the basics of machine learning. I understood in depth why and how important it is for the data to be tagged correctly, I was exposed to a major problem in the industry, an unsolvable problem. The topic made me go deep into the algorithms I learned. Also, I have being 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 me exposure to an interesting and contemporary world.