A reason of wrong prediction in image classification and a method to measure the effect of noise labels

Xinbin Zhang
Sydney Machine Learning Study Group
abner.zhangxinbin.anz@gmail.com

Abstract

How can we explain the wrong predictions of a machine learning algorithm? I develop a method using majority vote to classify the noise levels of the dataset and do some experiments on the MNIST dataset by manipulating different noise levels of the training dataset and test dataset. The result shows that that one single mislabeled instance in training dataset could cause the algorithm to make wrong prediction as mislead effect on clean test dataset. As number of noise instance increase, the mislead effect could be affected by other noise labels or be neutralized that the instance is predicted correctly again. And there are also compounded effect that wrongly predicted doesn't come from any group of noise instance but from the combined of all groups. When noise instances in training dataset increase, the accuracy on clean test dataset decreases and the accuracy on high noise level test dataset increases.

1 Introduction

When I start the learning of machine learning, I try different algorithms and models on MNIST dataset and I have a question "Why an algorithm more accurate than human makes wrong predictions on some good handwritings which are easy for human to judge?"

[Liang *et al.*, 2017] uses the influence function to explain "Why did the system make this prediction?" by looking at how it was derived from its training data to measure the effect of local changes.

[Ribeiro *et al.*, 2016] works on interpreting these black-box models by focus on understanding how a fixed model leads to particular predictions, e.g., by locally fitting a simpler model around the test point, which explains the predictions in terms of the model.

[Zhang *et al.*, 2017] demonstrate that deep networks are capable of memorizing the entire data even on corrupted labels. He shows that simple depth two neural networks already have perfect finite sample expressivity as soon as the number of parameters exceeds the number of data points as it usually does in practice.

These approaches help us to understand how the black box works. However, we still don't know why these black boxes make wrong prediction for easy instance after they achieving a very high accuracy.

In this paper, I try to answer this question by measuring the effect of noise labels. These effects cause the algorithms make wrong prediction and to achieve a higher accuracy on whole test dataset makes them predict easy instance wrong.

2 Related Work

In [Frenay *et al.*, 2014] shows that label noise has many potential negative consequences. For example, the accuracy of predictions may decrease, whereas the complexity of inferred models and the number of necessary training samples may increase.

There are several studies of the consequences of label noise on prediction performances.

Consequences worsen when differences in covariance matrices or misclassification rates increase. [Michalek *et al.*, 2017] shows that label noise affects normal discriminant and logistic regression: their error rates are increased and their parameters are biased.

[Okamoto *et al.*, 1997] present an average-case analysis of the kNN classifier. Classification performances are also affected by label noise.

[Nettleton *et al.*, 2010] compare the impact of label noise on four different supervised learners: naive Bayes, decision trees induced by C4.5, kNNs, and support vector machines (SVMs). The poor results of SVMs are attributed to its reliance on support vectors and the feature interdependence assumption.

[Weiss *et al.*, 1995] explains that small disjuncts (which individually cover only a few examples) are more likely to be affected by label noise than large disjuncts covering more instances. However, only large levels of label noise may actually be a problem.

[Brodley *et al.*, 1999] show that removing mislabeled samples reduces the complexity of SVMs (number of support vectors), decision trees induced by C4.5 (size of trees) and rule-based classifiers induced by RIPPER (number of rules). Post pruning also seems to reduce the consequences of label noise. Noise reduction can therefore produce models that are easier to understand, what is desirable in many circumstances.

[Biggio *et al.*, 2012] explores the training-set attack in the context of SVMs, [Mei *et al.*, 2015] extend the framework and applying it to linear and logistic regression.
In this paper, I extend the training-set attack to the deep learning to show how to measure the effect of noise label.

3. The method

3.1 The general procedure

Compared with related works that adds noise label data into training dataset, I use the method that normally is used in noise label cleansing to split the training dataset into different noise levels. The first step is to split data set into n parts and then identify candidate instances' noise level by using filter algorithms to count the times of incorrectly prediction. For each of the n parts, the filter algorithms are trained on the 1 part. Then using the filter to predict all instances. Each instance will be predicted n times and calculate the times of wrong predicted as the noise level of the instance. Instances with high noise level are more likely to be mislabeled instances.
In this way, I create the noise level 0 training dataset and noise level 0 test dataset, then use noise level 0 training dataset to train a ensemble network and test on noise level 0 test dataset and use the result as the benchmark, then I add one single noise instance or a group of different noise levels instances into the noise level 0 training dataset, and compare the result with the benchmark to measure the effect of the noise instances.

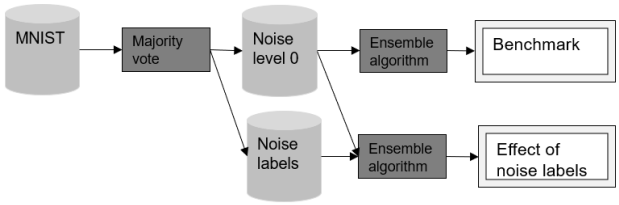

Figure. 1. The procedure of the method

3.2 The majority vote to classify noise level

The procedure is descripted as Figure2

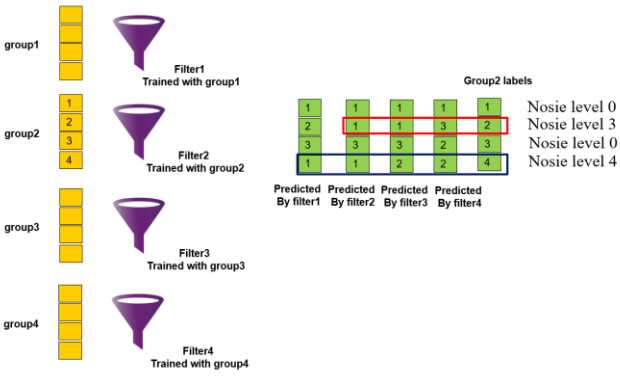

Figure. 2. shows how to classify instances into different noise levels

The base algorithm

The base algorithm(M1) is a 4 layers convolutional neural networks (CNNs) including:
First convolutional lay with filter size 4*4, channels 4;
Second convolutional lay with filter size 4*4, channels 16;
Third layer full connected size 2048;
The irritation is 10000;
The performance is 92.5%-95.5%.

The dataset splits

I split 60000 MNIST training data and 10000 test data into 7 groups, each group has 10000 digits. I use each group as training data to train a CNN as a filter, which has 91%-95% accuracy, then use the filter to predict all 7 groups and count the times of the wrong predicted as noise level for that instance.
D the combined data from training dataset Dtraining and test dataset Dtest
d = (x,y) one instance of D, x is the feature, in the image classification x is the picture, y is the label of x
D1 is the group1, which consists of 0-9999 instances of D.
D2 is the group 2, which consists of 10000-19999 instances.
D3 is the group 3, which consists of 20000-29999 instances.
D4 is the group 4, which consists of 30000-39999 instances.
D5 is the group 5, which consists of 40000-49999 instances.
D6 is the group 6, which consists of 50000-59999 instances.
D7 is the group 7, which consists of 60000-69999 instances.
Dn0 is noise level 0 in training dataset which is combined by instances that are predicted correctly by all base filters d=(x,y) that F(M1,D1,x) = y and F(M1,D2,x) = y and F(M1,D3,x) = y and F(M1,D4,x) = y and F(M1,D5,x) = y and F(M1,D6,x) = y and F(M1,D7,x) = y and d is in the 0-59999 of D.
Dn0' is noise level 0 in test dataset which is combined by instances that are predicted correctly by all base filters d=(x,y) that F(M1,D1,x) = y and F(M1,D2,x) = y and F(M1,D3,x) = y and F(M1,D4,x) = y and F(M1,D5,x) = y and F(M1,D6,x) = y and F(M1,D7,x) = y and d is in the 60000-69999 of D.
Dn1 is noise level 1 in training dataset which is combined by instances that are predicted correctly by 6 base filters, d=(x,y) that one of F(M1,Dj,x) != y, and d is in the 0-59999 of D, the noise level is 1.Dn1' is the combined instances d=(x,y) that one of F(M1,Dj,x) != y, and d is in the 60000-69999 of D.
Similarly, I get Dn2 to Dn7 and Dn7' to Dn7'.
The instances in Dn0 and Dn0' are coherent since they are predicted by the all base filters correctly.
Mislabeled instances will cause the incoherent between the instances and then affect the CNNs trained with these instances. But not all the instances in high noise level group are mislabeled, for example different style of handwriting in MNIST would be classified as high noise level.
The following tables show the count of different noise levels in MNIST training dataset and test dataset.
Table1 the counts of different noise levels in Dtraining

noise level	count
Dn0	52954
Dn1	1565
Dn2	961
Dn3	732
Dn4	635

Dn5	646
Dn6	712
Dn7	1795

Table2 the counts of different noise levels in Dtest

noise level	count
Dn0’	8930
Dn1’	217
Dn2’	159
Dn3’	123
Dn4’	107
Dn5’	85
Dn6’	103
Dn7’	276

Table3 show the distribution of each number in different noise level groups

	Dn0	Dn1	Dn2	Dn3	Dn4	Dn5	Dn6	Dn7	sum
0	5594	81	51	28	32	39	25	73	5923
1	6509	34	26	18	19	28	14	94	6742
2	5177	170	117	78	75	65	85	191	5958
3	5141	217	117	98	77	94	110	277	6131
4	5104	173	94	76	74	63	59	199	5842
5	4526	206	136	113	95	70	81	194	5421
6	5539	85	51	57	38	35	36	77	5918
7	5644	155	66	74	51	44	72	159	6265
8	4754	252	158	108	96	117	131	235	5851
9	4966	192	145	82	78	91	99	296	5949
sum	52954	1565	961	732	635	646	712	1795	60000

Table4 show the distribution of each number in different noise level groups

	Dn0'	Dn1'	Dn2'	Dn3'	Dn4'	Dn5'	Dn6'	Dn7'	sum
0	945	12	4	4	3	1	2	9	980
1	1107	2	7	4	3	1	2	9	1135
2	911	24	16	17	14	11	12	27	1032
3	888	29	13	17	14	7	10	32	1010
4	868	30	18	13	4	5	14	30	982
5	751	33	24	12	19	9	10	34	892
6	883	16	18	5	9	4	5	18	958
7	911	20	18	15	7	11	17	29	1028
8	810	23	22	26	15	20	18	40	974
9	856	28	19	10	19	16	13	48	1009
sum	8930	217	159	123	107	85	103	276	10000

3.3 The ensemble algorithm

The ensemble algorithm (M2) which is a 4 layers CNN including:
First convolutional lay with filter size 4*4, channels 4;
Second convolutional lay with filter size 4*4, channels 16;
Third layer full connected size 2048;
The irritation is 50000;
The ensemble is training and predicting 5 times and prediction is the digit that appears most times.
For all d=(x,y) belong to Dn0’, F(M2,Dn0,x)=y.
I use Dn0 as a clean training data and Dn0’ as a clean test data and achieve 100% accuracy.

3.4 The misled effect of single noise label on clean dataset

I add one mislabeled instance into the training data and measure the effect of noise instance by observing the result of prediction.
The d2080, d30310, d44960 and d54915 belong to Dn7 and are mislabeled instances.

 x2080, y2080 is 3

 x30310, y30310 is 5

 x44960, y44960 is 6

 x54915, y54915 is 4

I add d2080 into the Dn0, trained and teste on Dn0’ and find that one single mislabeled instance could mislead the prediction:

 x61466, y61466 is 5

 x68553, y68553 is 5

F(M2, Dn0+d2080,x61466)=3 while F(M2,Dn0,x61466)=5
F(M2, Dn0+d2080,x68553)=3 while F(M2,Dn0,x68553)=5
Because x2080 is 5 but being mislabeled as 3, adding this noise causes the 100% accuracy trained model to predict x61466 and x68553 as 3 rather than 5.
No instances in D0’ is affected by d30310 and d44960.

 x61941, y61941 is 7

F(M2, D0+d54915,x61941) = 4 while F(M2,D0,x61941) =7
Because x54915 is 7 but being mislabeled as 4, adding this noise causes the 100% accuracy trained model to predict x61941 as 4 rather than 7.
The difference results of mislabeled instances could be the limited instance of Dn0’, I add d2080, d30310, d44960 and d54915 into Dn0’ and draw the t-sne picture as below

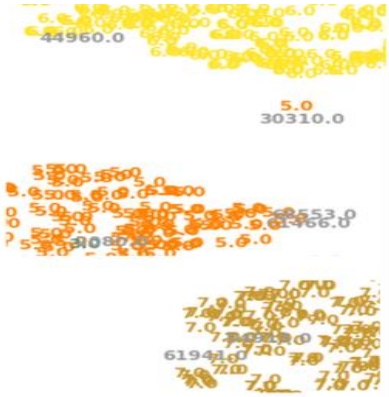

Figure3 shows that x2080 and x54915 are in the inside of 5 and 7, there are similar data such as x61466, x68553 and x61941 around them, while x30310 and x 44960 are not in the inside of 5 and 6.

When I remove d2080 and d54915 from Dn0 or correct the labels, no instance is affected.
Whether there are instances mislead in the test dataset is also decided by the test dataset. If I remove d61466, d68553 and d61941 from Dn0', no other instance is affected.

3.5 The effect of different noise levels data on clean test dataset

I add different noise level training instance into Dn0, and test on Dn0', and the number of incorrectly predicted instances as below table

Table5 show the count of wrong prediction of each level noise groups on Dn0'

Training dataset	Count of wrong prediction in Dn0'
Dn0	0
Dn0+n1	0
Dn0+n2	0
Dn0+n3	1
Dn0+n4	1
Dn0+n5	3
Dn0+n6	5
Dn0+n7	15
Dtraining	16

It shows that as the noise level increases there are more instances in Dn0' are predicted wrong. I analyze these wrongly predicted instances showed in appendix and find that there are misled effect, neutralized effect and compounded effect:

The neutralized effect

When only one mislabeled instance d2080, it misleads the prediction of x61466 and x68553. After adding another 1794 noise instances in Dn7 into training data, x61466 and x68553 are correctly predicted as 5:

F(M2, Dn0,x61466)=5
F(M2, Dn0+d2080,x61466)=3, misled effect
F(M2, Dn0+n7,x61466) = 5, neutralized effect
F(M2, Dn0,x68553)=5
F(M2, Dn0+d2080,x68553)=3, misled effect
F(M2, Dn0+n7,x68553) = 5, neutralized effect

The Misled effect

When there is only one mislabeled instance d54915, it misleads the x61941 to 4. Adding more noise instances into training data, x61941 is predicted as 8

F(M2, D0,x61941) =7
F(M2, D0+d54915,x61941) = 4 , misled effect
F(M2, Dtraining,x61941) = 8, misled effect

The compounded effect

When compare the wrong predicted instances of Dtraining, I find the compounded effect.

d60115 is predicted correctly from D0 to D0+7, but predicted wrong in Dtraining.

F(M2, Dn0+n1 ,x60115)=y60115
F(M2, Dn0+n2 ,x60115)=y60115
F(M2, Dn0+n3 ,x60115)=y60115
F(M2, Dn0+n4 ,x60115)=y60115
F(M2, Dn0+n5 ,x60115)=y60115
F(M2, Dn0+n6 ,x60115)=y60115
F(M2, Dn0+n7 ,x60115)=y60115
compounded effect:
F(M2,Dn0+n1+n2+n3+n4+n5+n6+n7 ,x60115)!=y60115

3.6 The effect of different noise levels data on noise test dataset

The misled effect, neutralized effect, and compounded effect are easily measured in clean test dataset D0'.

The misled effect, neutralized effect and compounded effect could happen at same time in high level noise test data. And the result of Dn7' shows that high level noise test data need high level noise training data to neutralize.

Table6 show the count of wrong prediction of each level noise groups on Dn0'

	Dn0'	Dn1'	Dn2'	Dn3'	Dn4'	Dn5'	Dn6'	Dn7'
Dn0	0	2	18	37	53	61	96	276
Dn0+n1	0	1	5	18	46	46	87	271
Dn0+n2	0	1	6	15	31	39	80	269
Dn0+n3	1	2	3	18	38	33	67	251
Dn0+n4	1	4	6	16	21	32	52	253
Dn0+n5	3	2	7	20	19	18	50	232
Dn0+n6	5	4	6	21	17	17	33	211
Dn0+n7	15	4	9	8	6	5	13	93
Dtraining	16	4	7	11	5	5	11	70

3.7 The reason of wrong prediction of good handwritings

To answer the question at the beginning of the paper. When the algorithm tries to reduce the loss, which is the wrong prediction on the whole test dataset, more high noise levels training dataset will make effect to reduce errors in Dn7', which has the cost of more instances in Dn0' being predict wrong.

Table7 show the count of wrong prediction of different training dataset

training dataset	count of predicted wrong in Dtest
Dn0	543
Dn0+n1	474
Dn0+n2	441
Dn0+n3	413
Dn0+n4	385
Dn0+n5	351
Dn0+n6	314
Dn0+n7	153
Dtraining	130

If I define the clean data Dn0 and Dn0' in MNIST as good handwriting, and Dn7 and Dn7' as bad handwriting. The consequence of predicting good handwriting wrong and predicting bad handwriting correctly is different. In real world, it is difficult to say what are correct labels for these bad handwritings, to predict them correct may have few

contributions. But predicting these good handwriting wrong would cause chaos when interacting with human.

4. Conclusions

In this work I do an experiment on MNIST and find that the noise labels could cause CNNs predict incorrectly, the misled effect could be explained by t-sne visualization, the neutralized effect and compounded effect may need more research. After classifying instances into different noise level, the effect is different that the accuracy on clean test dataset decreases when noise instances in training dataset increase and the accuracy on high noise level test dataset increases when noise instances in training dataset increase. And consequence of predicting clean instance and noise instance is different. Basing on the target of question to choose different noise level of training dataset.

The appendix

instance id	predicted wrong by dataset	effect
60018	Dn5	mislead effect, neutralized effect
60115	Dtraining	compounded effect
60184	Dn7	mislead effect, neutralized effect
60583	Dn7	mislead effect
60583	Dtraining	mislead effect
61941	Dn6	mislead effect
61941	Dn7	mislead effect
61941	Dtraining	mislead effect
62018	Dn7	mislead effect
62018	Dtraining	mislead effect
62414	Dn7	mislead effect
62414	Dtraining	mislead effect
62454	Dn7	mislead effect, neutralized effect
62939	Dn6	mislead effect
62939	Dn7	mislead effect
62939	Dtraining	mislead effect
62952	Dn6	mislead effect, neutralized effect
63023	Dtraining	compounded effect
63030	Dn5	mislead effect
63030	Dn6	mislead effect
63030	Dn7	mislead effect
63030	Dtraining	mislead effect
64400	Dn7	mislead effect, neutralized effect
64443	Dtraining	compounded effect
65199	Dtraining	compounded effect
66011	Dtraining	compounded effect
66024	Dn7	mislead effect, neutralized effect
66045	Dn7	mislead effect, neutralized effect
66532	Dn7	mislead effect
66532	Dtraining	mislead effect
68059	Dn4	mislead effect, neutralized effect
68325	Dn3	mislead effect
68325	Dn6	mislead effect
68325	Dn7	mislead effect
68325	Dtraining	mislead effect
68527	Dn7	mislead effect
68527	Dtraining	mislead effect
69158	Dtraining	compounded effect
69669	Dtraining	compounded effect
69792	Dn7	mislead effect, neutralized effect
69858	Dn5	mislead effect, neutralized effect

References

[1] [Brodley et al., 1999] C. E. Brodley and M. A. Friedl, “Identifying mislabeled training data,” J. Artif. Intell. Res., vol. 11, pp. 131–167, Jun. 1999.

[2] [Frenay et al., 2014] Classification in the Presence of Label Noise: a Survey, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 25, NO. 5, MAY 2014

[3] [Nettleton et al., 2010]D. Nettleton, A. Orriols-Puig, and A. Fornells, “A study of the effect of different types of noise on the precision of supervised learning techniques,” Artif. Intell. Rev., vol. 33, no. 4, pp. 275–306, 2010.

[4] [Weiss et al., 1995] G. M. Weiss, “Learning with rare cases and small disjuncts,” in Proc. 12th Int. Conf. Mach. Learn., Tahoe City, CA, USA, Jul. 1995, pp. 558–565.

[5] [Michalek et al., 1980]J. E. Michalek and R. C. Tripathi, “The effect of errors in diagnosis and measurement on the estimation of the probability of an event,” J. Amer. Statist. Assoc., vol. 75, no. 371, pp. 713–721, 1980.

[6] [Okamoto et al., 1997]S. Okamoto and Y. Nobuhiro, “An average-case analysis of the k-nearest neighbor classifier for noisy domains,” in Proc. 15th Int. Joint Conf. Artif. Intell., vol. 1. Nagoya, Japan, Aug. 1997, pp. 238–243.

[7] [Biggio et al., 2012] Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In International Conference on Machine Learning (ICML), pp. 1467–1474, 2012.

[8] [Mei et al., 2015]Mei, S. and Zhu, X. Using machine teaching to identify optimal training-set attacks on machine learners. In Association for the Advancement of Artificial Intelligence (AAAI), 2015b.

[9] [Liang et al., 2017] Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." arXiv preprint arXiv:1703.04730 (2017).

[10] [Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. ”why should I trust you?”: Explaining the predictions of any classifier. In International Conference on Knowledge Discovery and Data Mining (KDD), 2016.

[11] [Zhang et al., 2017] Zhang, Chiyuan, et al. "Understanding deep learning requires rethinking generalization." arXiv preprint arXiv:1611.03530 (2016).