Handling Label Noises in Dataset

CSE 5522 Group8

Ziming Ma, Zhiyuan Zhao, Xiaochen Zong

Why this matters

Datasets are paramount for training machine learning models. However, in real life, datasets clean of noisy labels are rare, and label noises caused by human error are quite common. Meanwhile, training with noises also illustrates some rules in machine learning. In this study, we focus on handling these noises, building robust and accurate models, as well as trying to interpret some noticeable facts.

Label Noises

In this study, we will focus on uniform noise.

Uniform Noise means, if the the set of total classes is S, then for training data X_train, its correct label being y, we intentionally label it as Y_noise, where Y_noise is uniformly distributed among S / { y }.

For the training data below, we label it x, where x is uniformly distributed among:

[0,1,3,4,5,6,7,8,9]

The affects of uniform label noises on learning

Generally, the larger proportion of noisy labels in training set, the worse learning outcome.

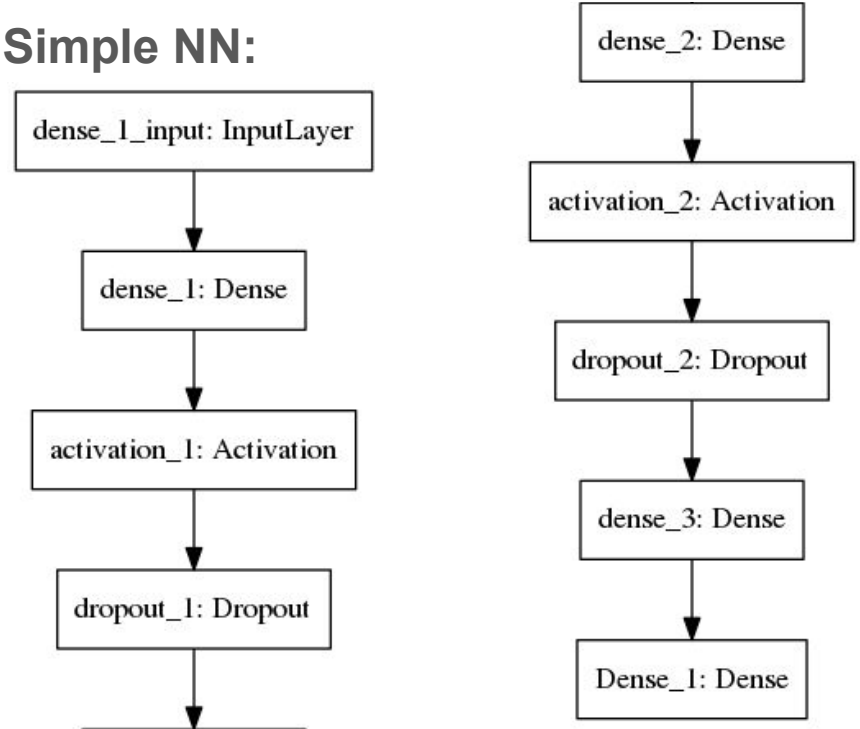
Meanwhile, simpler models are usually more robust to noises than complex ones(Simple Neural Network vs. AlexNet).

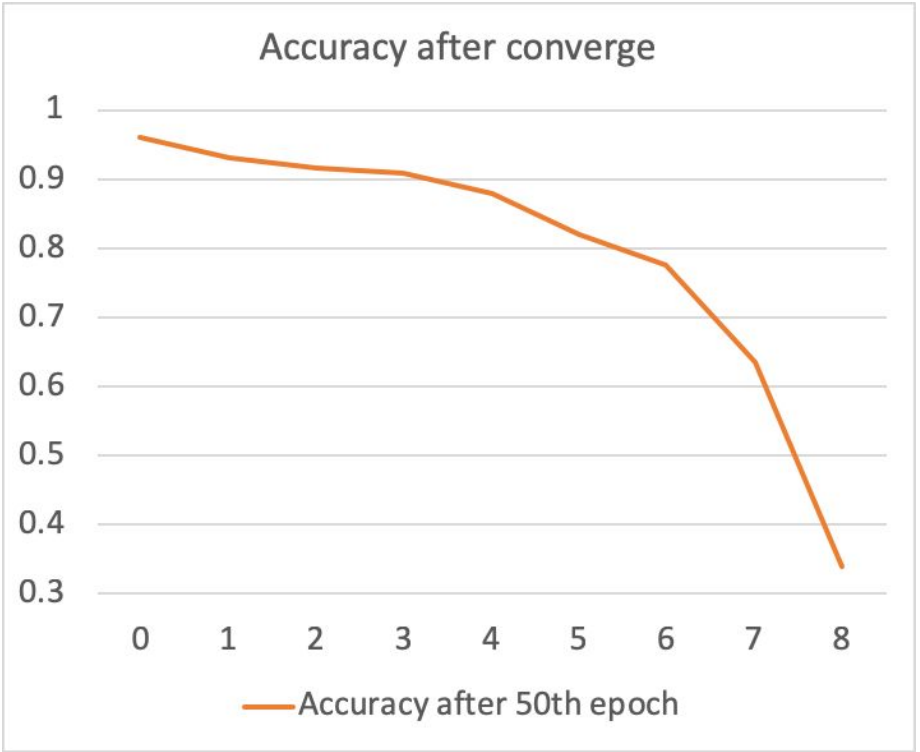
This is because complex models tend to **overfit**, i.e. learn details in noisy examples and distinguish them from correct examples, while simpler models have better generalization ability. We'll talk about this later.

The affects of uniform label noises on learning

For noisy label added exogenously, i.e., the original training set is not corrputed:

Simple NN:

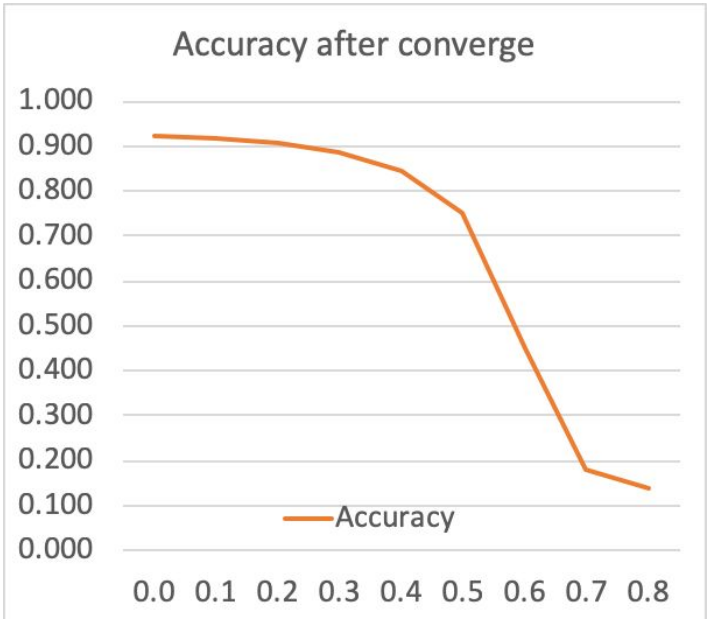




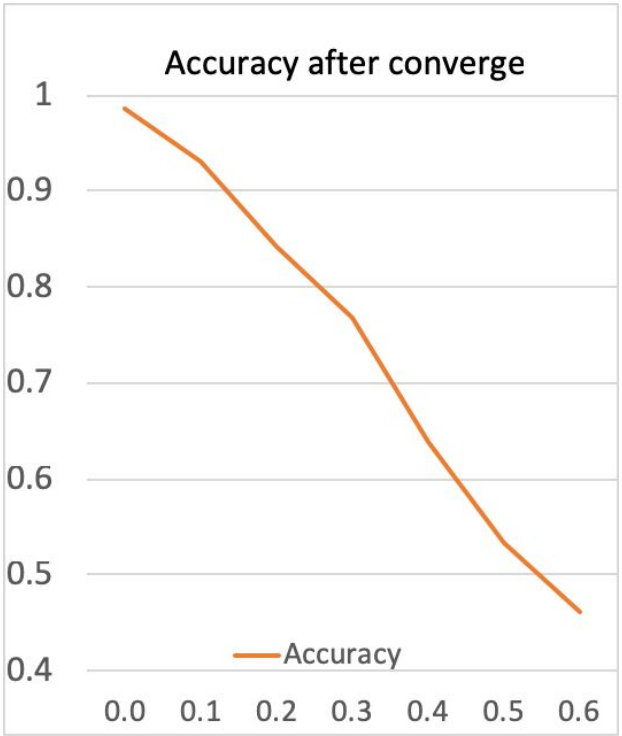
The affects of label noises on learning

For noise added by corrupting training data labels:

Simple NN:

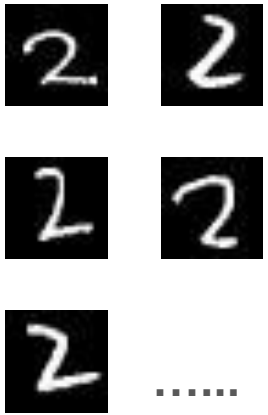


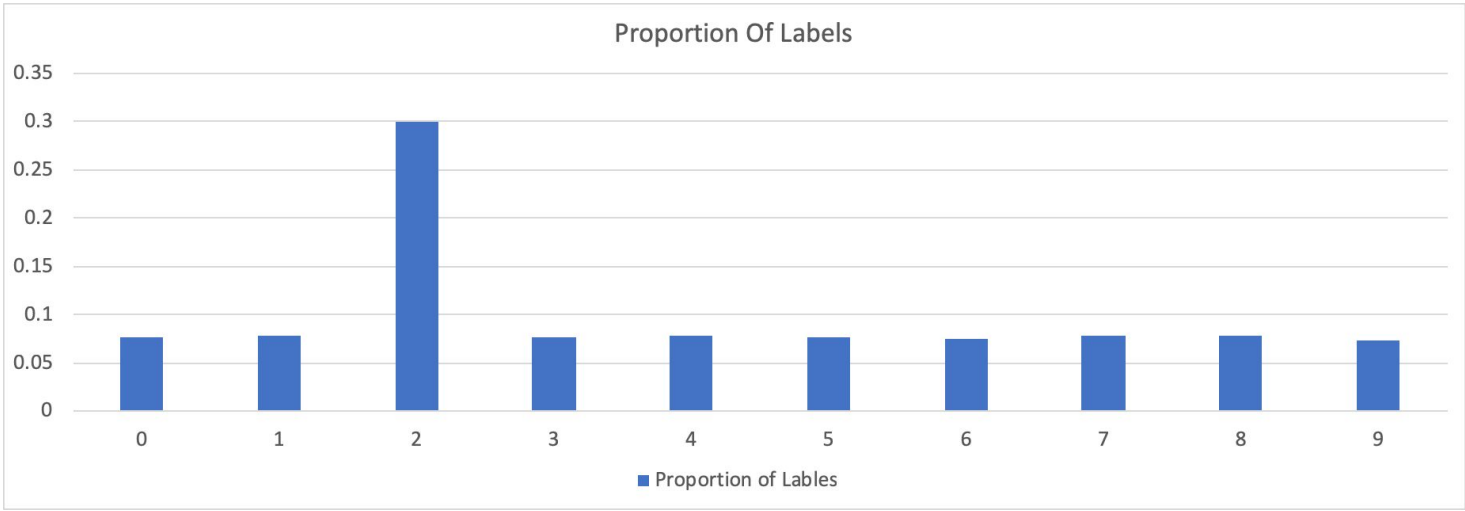
AlexNet:



Why high accuracy is still possible with large proportion of noisy labels?

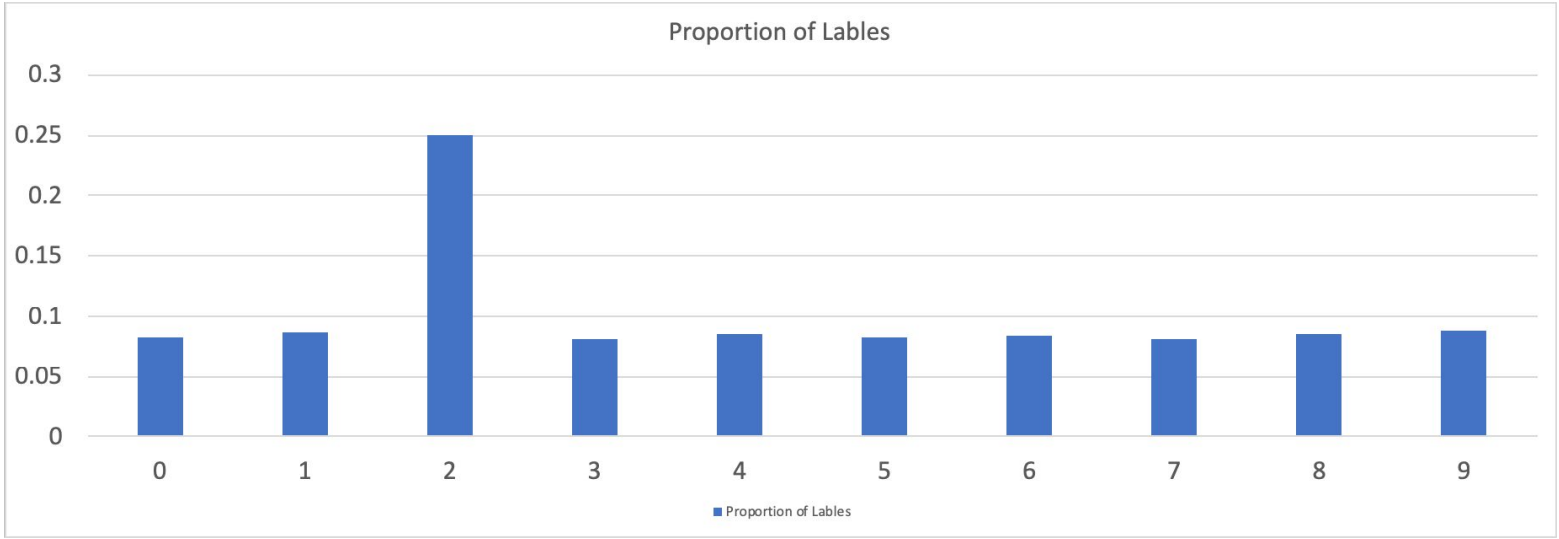
When noisy labels take up 30% of training data, for a particular kind of pictures, e.g. pictures with character “2” on it, the distribution of their labels is like the follow:





Why high accuracy is still possible with large proportion of noisy labels?

When noise takes up 300% of the training data on a particular kind of picture, it still has high accuracy.



Why high accuracy is still possible with large proportion of noisy labels?

$$H_s = \sum_{i=1}^{n} p_i I_e = -\sum_{i=1}^{n} p_i \log_2 p_i$$

It can be proved that, to maximize information entropy, H, we should include as many classes in the dataset as possible, while keeping the distribution of each label uniform. MNIST achieves this.

In MNIST, there are ten possible outcomes for one picture. The more outcomes, the lower proportion of incorrect examples in one label. This works like “diluting” incorrect labels. During training, these false labels would “cancel out” each other.

Attempts to Attack Label Noises

- Use Simple Nerual Network as Pre-trained model to identify false labels.
- Use Simple Nerual Network to find ambiguous training pictures.
- Utilize SVM to clean mislabeled training data.
- Configurate hyper-parameters and stop training early to prevent overfitting on complex models(e.g., AlexNet).

Attempts to Attack Label Noises

Use Simple Nerual Network as Pre-trained model to identify false labels.

Experimental setup:

Assume we have MNIST, but 20% of the total 70000 examples are corrputed by uniform label noise.

We train Simple NN on training set of 60000, reaching 90.6% accuracy.

Then we use it to predict examples in test set of 10000, in which 2000 are noises. For those have different labels from our prediction, we regard them as noisy labels and pick them out.

Attempts to Attack Label Noises

Use Simple Nerual Network as Pre-trained model to identify false labels.

Result:

2613 examples are picked out from 10000.

Selected examples:

Labels: 9 1 9 4 7 4 2 7 8

Attempts to Attack Label Noises

Use Simple Nerual Network to find ambiguous training pictures.

Experimental setup:

From the trained Simple NN in last experiments, we get its prediction of probabilities of each labels(0~9) for the entire training set of 60000. If a example has the largest probability < 0.38 while the second largest > 0.1, we regard it as ambiguous.

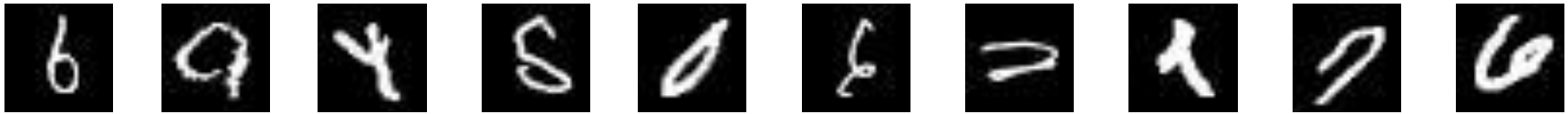
Attempts to Attack Label Noises

Use Simple Nerual Network to find ambiguous training pictures.

Result:

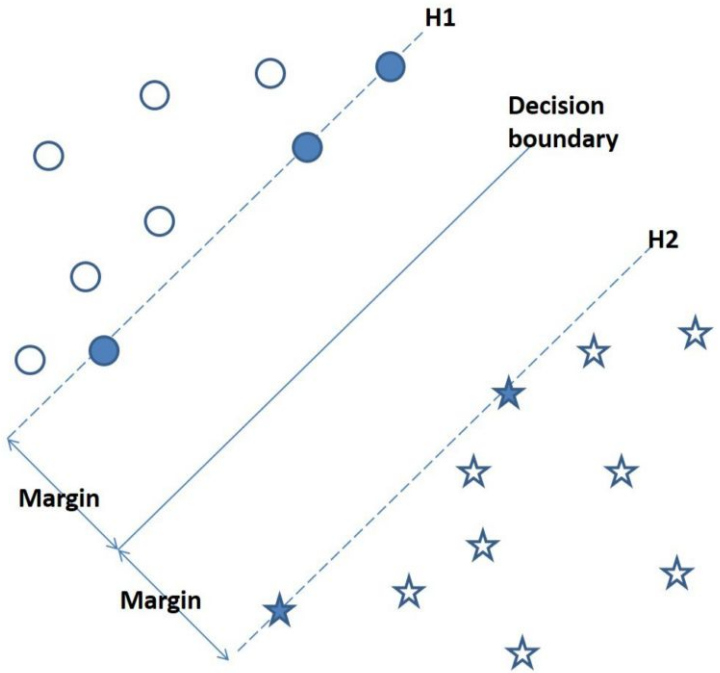
130 example were picked out.

Selected examples:



Utilize SVM to Clean Noisy Label

Assumption: The mislabeled examples tend to be on the margin and are more likely to be chosen as support vectors of the SVM classifier



Dataset

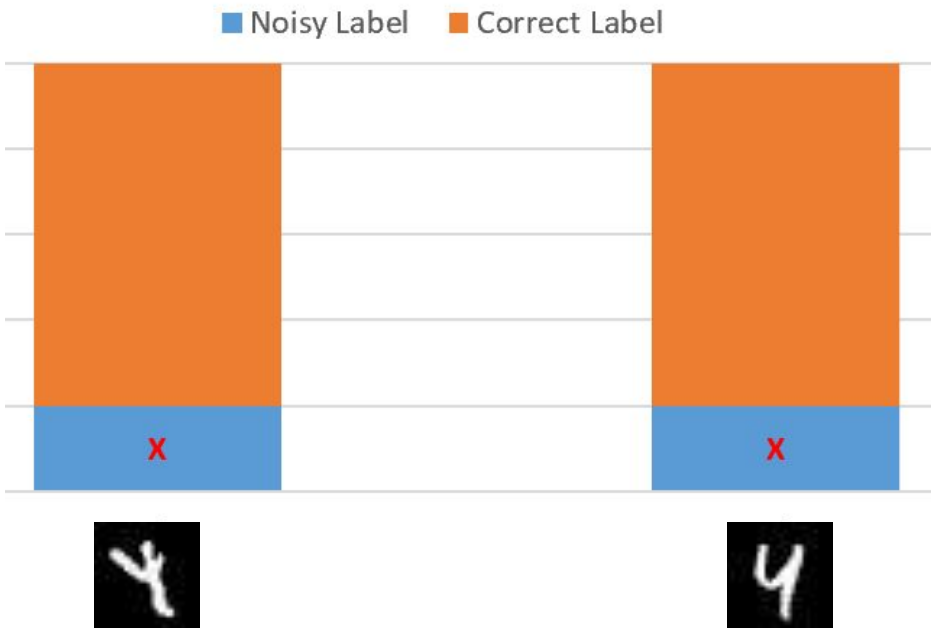
MNIST

2000 samples from

each label (totally 4000)

X label is flipped

X = 10%, 20%, 30%



Clean Noisy Label by Two-class SVM

Algorithm:

1. Train a binary SVM classifier using the training examples.
2. Collect and review all the support vectors .
3. Correct the labels of the mislabeled examples in the support vectors.
4. Repeat Steps 1-3 until no further noisy label is found.

class sklearn.svm.SVC()

Result

X = 10%(400 noise)

C = 8

kernel = RBF

gamma = 1/ (# features)

#· Iteration¤	%· Cumulative· #SV¤	%·Cumulative·#· Noisy·label· corrected¤	#·Noisy· label· corrected¤	#·SV¤
1¤	41.30%¤	99.50%¤	398¤	1652¤
2¤	41.90%¤	100%¤	2¤	640¤
3¤	41.90%¤	100%¤	0¤	640¤

Result

X = 20%(800 noise)

#· Iteration¤	%· Cumulative· #SV¤	%·Cumulative·#· Noisy·label· corrected¤	#·Noisy· label· corrected¤	#·SV¤
1¤	61.65%¤	97.5%¤	780¤	2466¤
2¤	62.12%¤	99%¤	792¤	576¤
3¤	62.2%¤	99.5%¤	796¤	563¤

X = 30%(1200 noise)

#· Iteration¤	%· Cumulative· #SV¤	%·Cumulative·#· Noisy·label· corrected¤	#·Noisy· label· corrected¤	#·SV¤
1¤	76.8%¤	97.0%¤	1164¤	3072¤
2¤	77.2%¤	98%¤	1176¤	668¤
3¤	77.25%¤	99%¤	1188¤	665¤

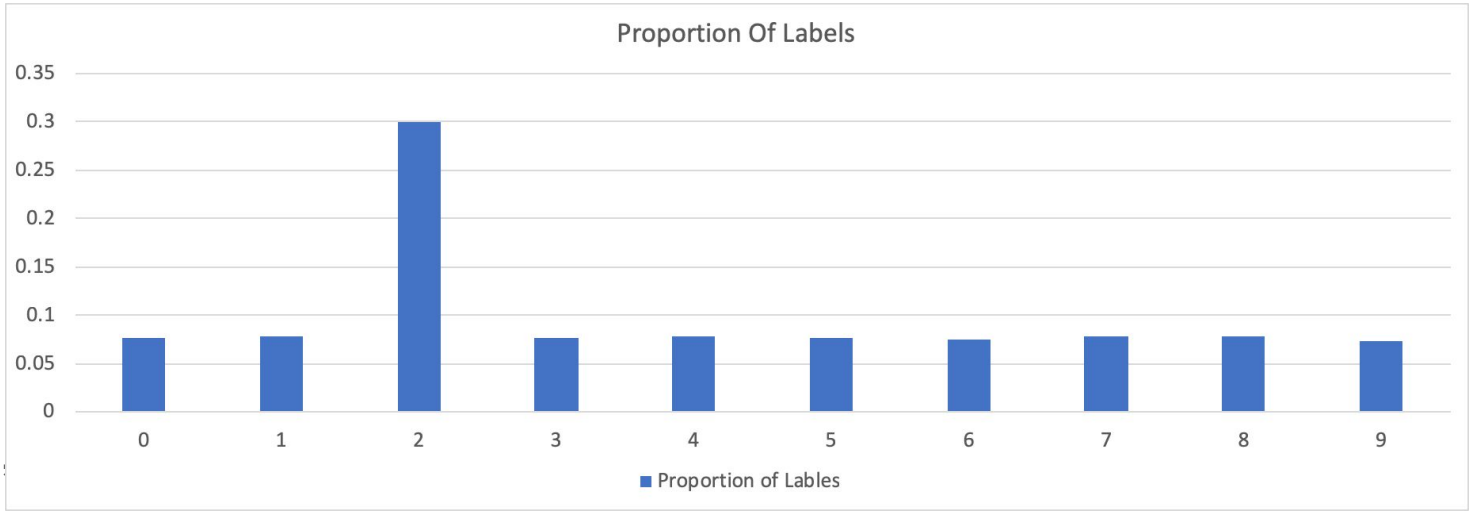
Dataset

MNIST

Uniform Label Noise

size = 3600

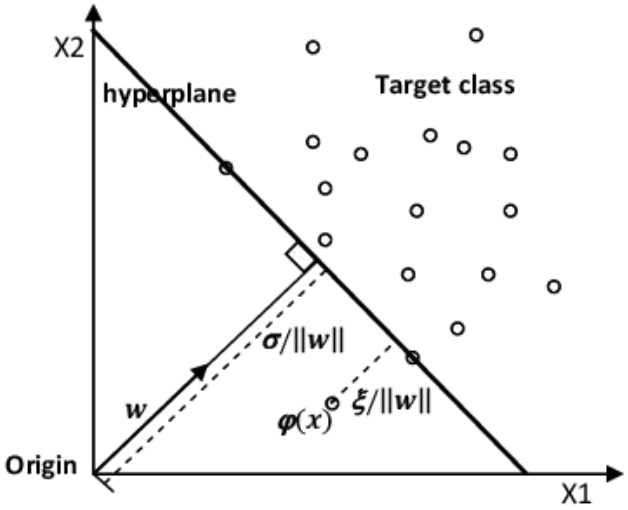
X = 10%, 20%, 30%



Clean Noisy Label by One-Class SVM

Algorithm:

1. Train OCSVM classifier using the training examples.
2. Collect and review all the outliers .
3. Remove mislabeled examples in the outliers.
4. Repeat Steps 1-3 until no further noisy label is found.



class sklearn.svm.OneClassSVM()

Result

X = 10% (360 noise)

μ = 0.5

kernel = RBF

gamma = 1/ # features

#· Iteration¤	%· Cumulative· #SV¤	%·Cumulative·#· Noisy·label· corrected¤	#·Noisy· label· corrected¤	#·SV¤
1¤	**50.06%¤**	**94.72%¤**	**341¤**	**1802¤**
2¤	72.47%¤	98.06%¤	353¤	1632¤
3¤	83%¤	98.33%¤	354¤	1627¤
4¤	83.36%¤	98.33%¤	354¤	1626¤
5¤	83.36%¤	98.33%¤	354¤	1626¤

X = 20% (720 noise)

μ¤	%· Cumulative· #SV¤	%·Cumulative·#· Noisy·label· corrected¤	#·Noisy· label· corrected¤	#·SV¤
0.5¤	50.03%¤	90.56%¤	652¤	1801¤
0.55¤	55.11%¤	92.92%¤	669¤	1984¤
0.6¤	60.03%¤	95.00%¤	684¤	2161¤

x = 30% (1080 noise)

μ¤	%· Cumulative· #SV¤	%·Cumulative·#· Noisy·label· corrected¤	#·Noisy· label· corrected¤	#·SV¤
0.6¤	60.03%¤	89.72%¤	969¤	2161¤
0.65¤	65.08%¤	92.22%¤	996¤	2343¤
0.7¤	70.08%¤	96.02%¤	1037¤	2523¤

Issue with reviewing

1. The SVM classifier has the property to capture mislabels examples as its support vector.
2. Relative small dataset with a small percentage of noisy label (around 10%).
3. Not very useful for large dataset or large percentage of noisy label (over 20%).

Attampt to get rid of reviewing

Algorithm:

1. Train OCSVM classifier using the training examples.
2. Remove all the outliers(SV) from the dataset
3. Train a new OCSVM classifier with the updated dataset
4. Use the new classifier to classify the removed outliers(SV).
5. Remove the ‘-1’ examples and keep the ‘1’ examples.
6. Repeat Steps 1-5 until no further noisy label is found.

Result

MNIST

Uniform Label Noise

size = 3600

X = 10% (360 noise)

#· iteration¤	%· Cumulative· Noisy·Label· Removed¤	%·Cumulative· Correct·Label· Removed¤	#·Noisy· Label· Removed¤	#·Correct· Label· Removed¤	Size·of· Dataset¤
1¤	40.0%¤	6.79%¤	144¤	220¤	3236¤
2¤	63.06%¤	14.32%¤	227¤	464¤	2912¤
3¤	73.61%¤	22.04%¤	265¤	714¤	2621¤
4¤	83.61%¤	29.07%¤	301¤	942¤	2357¤
5¤	89.44%¤	35.77%¤	322¤	1159¤	2119¤
6¤	94.72%¤	41.79%¤	341¤	1354¤	1905¤

Attempts to Attack Label Noises

Configurate hyper-parameters and stop training early to prevent overfitting on complex models(e.g., AlexNet).

Experimental setup:

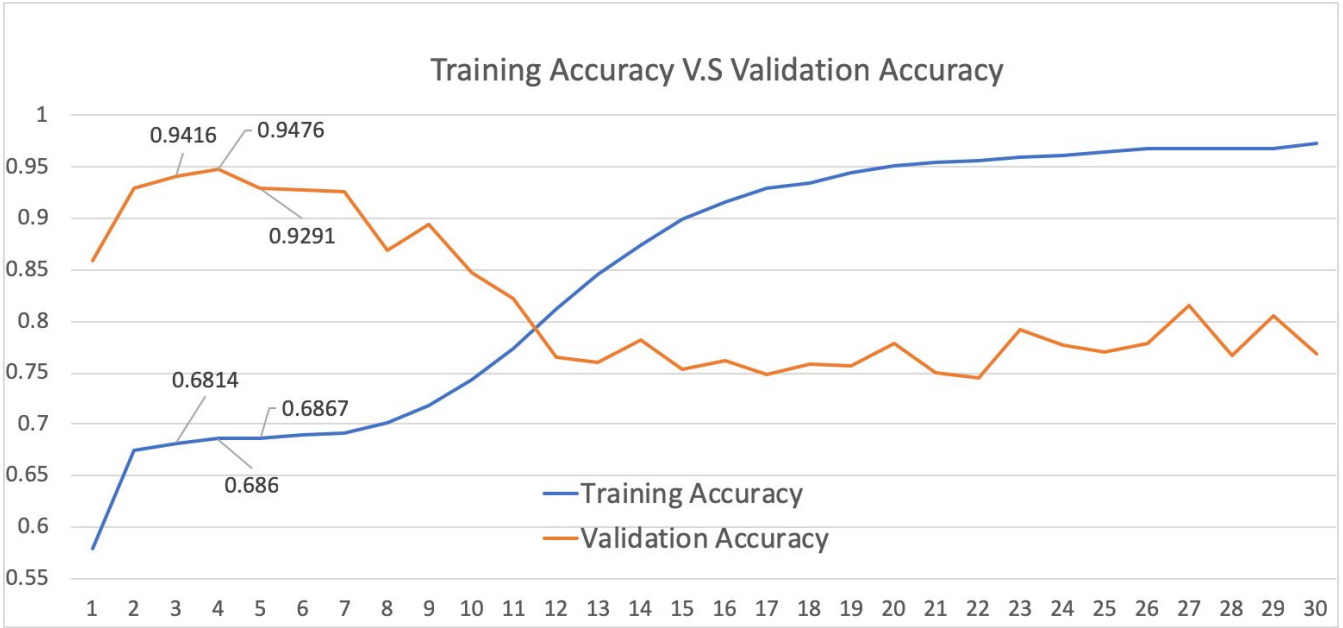
Noisy labels take up 30% of training data, other 70% are correct labels.

For the last two dense layers of ALexNet, we set them to have 500 neurons, with dropout = 0.5(same as Simple NN)

Attempts to Attack Label Noises

We try to stop training at the very time when:

1. Validation accuracy reaches highest. Or:
2. Proportion of correct examples × Validation accuracy = Training Accuracy.



Future work

An iteration procedure to clean noisy labels while building highly accurate complex models: Split the entire training set evenly into into two parts, A and B. Train AlexNet on A and stop training at the right time to gain highest validation accuracy, then use this model to clean B; then train a new AlexNet on cleaned B to gain highest validation accuracy, and use it to clean A, and so on. Finally, after a few iteration, we will get a pretty clean dataset as well as a highly accurate AlexNet.

Reference

1. Amnon Drory, Shai Avidan and Raja Giryes, “On the Resistance of Neural Nets to Label Noise”, arXiv:1803.11410 [cs.LG], 30 Mar 2018.

2. Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht and Oriol Vinyals, “Understanding deep learning requires rethinking generalization”, arXiv:1611.03530 [cs.LG], 26 Feb 2017.

3. David Rolnick, Andreas Veit, Serge Belongie and Nir Shavit, “Deep Learning is Robust to Massive Label Noise”, arXiv:1705.10694 [cs.LG], 26 Feb 2018.

4. Frédéric Branchaud-Charron, Fariz Rahman, Fariz Rahman, Taehoon Lee, Keras: Deep Learning for humans, https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py

5. Feisiqi, Into to Keras: Hello Keras on MNIST-Handwritting Recognition, https://blog.csdn.net/sdust_dx/article/details/80365674

6. N. Natarajan I. Dhillon P. Ravikumar A. Tewari "Learning with noisy labels" Proc. Neural Inf. Process. Syst. pp. 1196-1204 2013.

7. Rajmadhan Ekambaram, “Label Noise Cleaning Using Support Vector Machines”, 26 Feb 2017.

8. Sergiy Fefilatyev, Matthew Shreve, Kurt Kramer, Lawrence Hall, Dmitry Goldgof, Rangachar Kasturi, Kendra Daly, Andrew Remsen, Horst Bunke, “Label-Noise Reduction with Support Vector Machines”, 15 Nov. 2012.

Reference

9. Yao Wang, Influence and recognition of noisy labels in deep learning, https://blog.csdn.net/wangyao_bupt/article/details/77485553