Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5514--20-10,039

Towards High Performance Network Training with Noisy Label Datasets

Leslie N. Smith
Elizabeth A. Gilmour

NCARAI Branch
Information Technology Division

April 10, 2020

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
04-10-2020	NRL Memorandum Report	

4. TITLE AND SUBTITLE

Towards High Performance Network Training with Noisy Label Datasets

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Leslie N. Smith and Elizabeth A. Gilmour

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
1J06

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5514--20-10,039

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Creating large amounts of labeled data to train neural networks is an obstacle to applying deep learning to new applications. Heuristic methods for labeling data typically produce a significant fraction of mislabeled samples. This report describes some methods in the literature that find the fraction of noisy labeled datasets that are probably labeled correctly and our efforts to improve on these methods. The method we describe and test is called "NoisyLabel Correcting Cross Validation". The results of this method proved inferior to the INCV method in the literature but the new understandings learned from this effort inspired two new methods: the generalized sensitivity analysis and the soft lables approaches. Our future plans include testing these methods.

15. SUBJECT TERMS

Deep learning
Automatic labeling
Training with noisy labeling
Supervised classification
Small dataset learning

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Leslie N. Smith
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	9	**19b. TELEPHONE NUMBER** *(include area code)* (202) 767-9532

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

Towards High Performance Network Training with Noisy Label Datasets

Elizabeth A. Gilmour and Leslie N. Smith
Naval Center for Applied Research in Artificial Intelligence
U.S. Naval Research Laboratory
Washington, D.C. 20375
{elizabeth.gilmour, leslie.smith}@nrl.navy.mil

December 8, 2019

Abstract

Creating large amounts of labeled data to train neural networks is an obstacle to applying deep learning to new applications. Heuristic methods for labeling data typically produce a significant fraction of mislabeled samples. This report describes some methods in the literature that find the fraction of noisy labeled datasets that are probably labeled correctly and our efforts to improve on these methods. The method we describe and test is called "Noisy Label Correcting Cross Validation". The results of this method proved inferior to the INCV method in the literature but the new understandings learned from this effort inspired two new methods: the generalization sensitivity analysis and soft labels approaches. Our future plans include testing these methods.

1 Introduction

Training deep neural networks for real-world models requires large amounts of labeled data. Labeling these datasets typically require humans to manually label from hundreds to millions of images. In some cases, the labels are general knowledge, but in other cases, such as for aerial imagery, correctly labeling the images requires specialized knowledge. Unfortunately, the need to create large labeled datasets are often the limiting factor on new applications of deep learning.

To alleviate the burdensome effort of manually labeling large quantities of training samples, some researchers are investigating automatic methods for labeling [10, 2]. Unfortunately, the results from these heuristic labeling methods to date contain a significant fraction of incorrect labels. Therefore, novel methods that detect and correct incorrect labels are needed.

Datasets with inaccurate labels for training samples are called noisy label datasets. Noisy labels reduce accuracy for deep learning models because deep neural networks are trained to match the training samples labels, whether they are right or wrong. It has been shown that even with random labels, neural networks can memorize the training data but, as expected, are unable to generalize to other data [14].

To the extent that the labels are correct, neural networks learn by incrementally modifying the trainable parameters (i.e., weights and biases) in the neural networks to reduce the loss of the models predictions on the training data. Some examples are relatively easy for the network to accurately classify and the network first learns these examples. As the network continues to iterate over the data, it later learns the more difficult examples [1]. By memorizing difficult examples, the neural network is trained to accurately predict the label for each training sample, even when the label is wrong. As a result, these incorrectly labeled samples can lower the accuracy of neural network on unseen samples [3]. Deep neural networks can fit even random labels, but in that case, they will not generalize

Manuscript approved Month 00, 2020.

well to new, unseen data [14].

New applications of machine learning will require large amounts of labeled data. Whether because of a lack of expertise on the part of the labelers or because of the use of automated labeling methods, some of this new data may contain too many errors to train neural networks to generalize well and make accurate predictions on unseen data. To clean these datasets, new methods to remove labeling errors will be necessary so they can be effectively used to train deep learning models.

In this report we describe a novel method to find and correct incorrectly labeled samples. Our method builds on the iterative noisy cross validation (INCV) method described in [3]. In Section 2 we describe related methods from the literature. In Section 3 we describe our method and our results are described in Section 4.

2 Related Work

Numerous methods have been proposed to deal with noisy labels. Several methods focus on estimating the noise transition matrix and correcting the objective function accordingly, such as forward or backward correction [9] and the S-model [4]. Using the predictions of DNNs provides another set of approaches to correct labels, such as Bootstrap [11], Joint Optimization [13] and D2L [7]. An alternative method is to train on weighted samples, as demonstrated by Decoupling [8], MentorNet [6], gradient-based reweighting (citeren2018learning and Co-teaching [5]. The primary issue is to design a reliable and convincing criteria of selecting or weighting samples.

Co-teaching [5] is a method for selecting samples from a dataset to create a cleaner but smaller dataset. It is based on the idea that clean samples will have a lower training loss than mislabeled samples. If the ratio of mislabeled samples to correctly labeled samples is known, the mislabeled samples can be discarded and the correctly labeled samples can be added to a selected set for training a classifier.

Co-teaching uses two neural networks. Each network is trained on half of the data. For each batch of data, the samples with the lowest training loss are thought to be clean samples and are used to "teach" the other network. This cleaned batch of data is passed to the other network which is trained with the cleaned batch of data and also selects a set of the cleanly labeled data. The ratio of clean to noisy samples is needed to determine how many samples to select.

Throughout this process, co-teaching sequentially adds clean samples with the lowest training loss to the selected set until it develops a cleaner dataset. This cleaner dataset can be used to train a reinitialized neural network. Co-teaching has shown success in training neural networks despite very noisy training data. It has reached a classification accuracy of 91% on the MNIST dataset and 74% on the Cifar-10 dataset (state-of-the-art performance is over 99% and 98%, respectively) with a symmetric noise ratio of 0.5 i.e., half of the samples had incorrect labels).

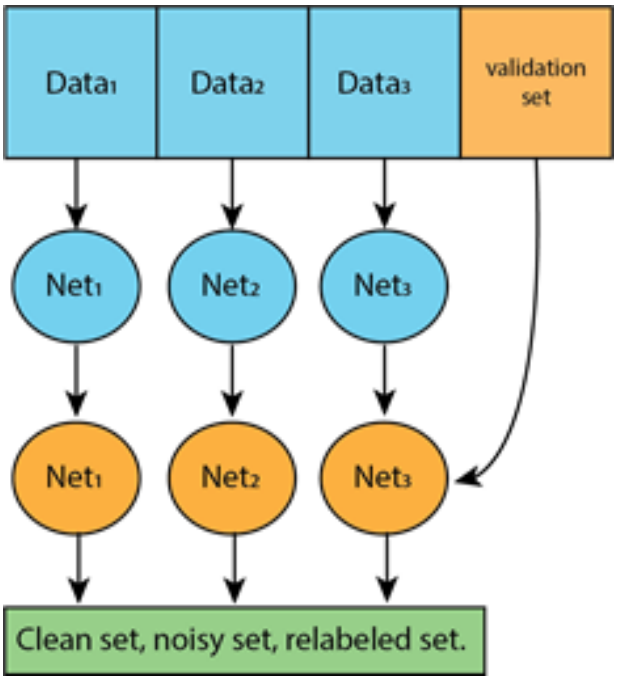

Figure 1: Diagram of the Noisy Label Correcting Cross Validation method

Iterative noisy cross validation (INCV) is a method that builds off of the work with co-teaching. INCV creates a set of clean labeled data so that the co-teaching method begins with a set of accurately labeled data and can more quickly clean the data and have a more stable training process ([3]). Additionally, unlike co-teaching, INCV

Table 1: Results of experiments using Resnet 110 for the architectures of all three networks.

Noise ratio (symmetric)	Relabeling iterations	Epochs per relabeling interation	Co-teaching training epochs	Test accuracy (%)
0.0	4	10	50	66
0.4	4	10	50	44

Table 2: Results from the tests with three different network architectures.

Noise ratio (symmetric)	Relabeling iterations	Epochs per relabeling interation	Co-teaching training epochs	Test accuracy (%)
0.0	4	10	50	69
0.4	4	10	50	55

does not require the noise ratio, so it can work with noisy data where the ratio of noisy labels to clean labels is not known, which is the typical situation.

In the INCV method, two neural networks are each trained on half of the data. The networks are then tested on the unseen half of the data. In the testing phase, samples are iteratively added to a clean set if the predicted label is the same as the given label. Iteratively, networks are trained on both the clean set and a subset of the noisy set until a clean set is created to start the co-teaching process. This process further improves upon co-teaching and leads to higher classification accuracy, even with very high noise ratios. However, co-teaching and INCV eliminates a large portion of correctly labeled training samples that are difficult for the networks to classify correctly. We attempted to improve on INCV, especially for these hard examples.

3 Methods

We proposed developing a method that builds off of co-teaching and INCV methods, both of which only select a subset of the training data that is deemed to have correct labels. The objective was to be able to obtain a larger number of training samples by not discarding mislabeled samples but to estimate the true labels automatically. In addition to building off of co-teaching and INCV methods, we utilize the results of [1] who showed that DNNs learn simple patterns quickly and later they tend to memorize the difficult patterns. Our proposed method uses voting by three different neural networks to not only identify clean samples, but also to fix the labels of noisy samples in order to produce a clean training set from a noisy set.

Our method, which we called "Noisy Label Correcting Cross Validation", starts by splitting the data into four subsets (see Figure 1). Three subsets of the data are used to each train one neural network. Each of the three neural networks is tested with the same validation subset.

Samples in the validation set are added to one of three datasets depending on voting between the three trained networks: the clean set, the relabeled set, or the noisy set. The samples in the validation subset are added to the clean set if all three networks predict the same label and the predicted label is the same as the given label. If all three networks make the same prediction but their prediction is not the same as the given label, the sample is relabeled with the predicted and added to the relabeled set. If the predicted labels vary among the three networks, the samples is added to the noisy set.

As with the INCV method, the cleaning and relabeling step is repeated. For each iteration, the three networks are re-initialized and then trained from scratch on a subset of the noisy labels, the entire re-labeled set, and the entire clean set. As a result, in each iteration the neural networks should be trained on a larger portion of clean data and a

smaller portion of noisy data. The goal was to develop a method that would be able to create a large clean set by relabeling noisy samples rather than simply assigning clean samples to the clean set as in INCV or co-teaching.

4 Results

4.1 Relabeling experiment

Tests were run using the Cifar-10 dataset, a dataset of 32x32x3 pixel images of objects belonging to ten common classes, including horses, dogs, cats, trucks, and boats. The labels were symmetrically switched on different portions of the data to introduce symmetric noise with different noise ratios as needed for experiments.

The training set is divided into four subsets. One subset is reserved for validation and three subsets are used to train three neural networks. Initially, the neural networks all used the Resnet 110 architecture. The networks were trained on the data for 50 epochs before testing on the reserved test subset. The validation data was added to the clean set, relabeled and added to the clean set, or left in the noisy set depending on the voting among the three neural networks.

In the next iteration, the subsets of training data were shuffled. Once again, three neural networks were initialized and trained on a quarter of the training data. In this and in further iterations, the networks were trained on the subset of training data as well as the clean and relabeled data. For all tests, the Noisy Label Correcting Cross validation method was run for four iterations. Unfortunately, the results of this method were inferior to the INCV method.

Due to the unsatisfactory results, we tested different network architectures for the three neural networks. We believed the three networks trained on different subsets of the data would not make the same mistakes. For subsequent tests, we used Resnet 32, Resnet 110, and Resnet 164 for the three neural networks (Table 2).

Using three different network architectures improved the test accuracy somewhat; the accuracy for a noise ratio of 0.4 increased by 11% over the previous test. With either the same network architecture or with different network architectures, the Noisy Label Correcting Cross Validation method does not perform as well as either co-teaching or INCV.

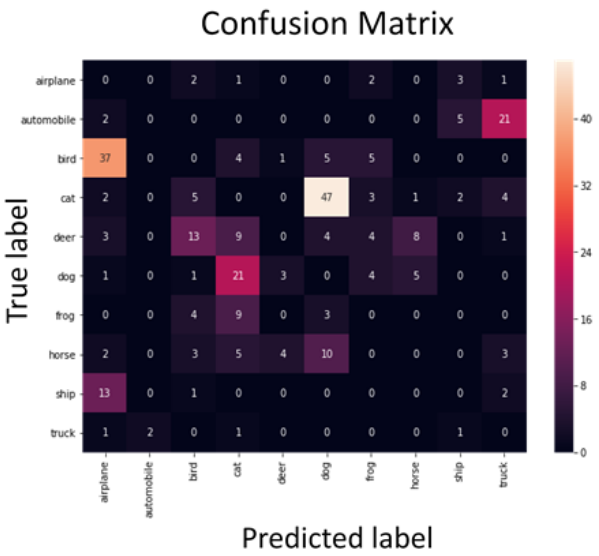

Figure 2: A confusion matrix of true and predicted labels for clean Cifar-10 data.

4.2 Understanding this failure: Confusion between classes

A major reason for the lack of success of Noisy Label Correcting Cross Validation method is the confusion between similar classes. The three networks were all in agreement for confusing classes and this led to mislabeling correctly labeled data. This was because the label would automatically be changed if all three networks agreed on the same label.

We had believed that it would be unlikely for all three networks to be in agreement about the wrong label, but surprisingly, they often were in agreement and incorrect for the hard training samples, such as distinguishing between a cat and a dog. In this case, the networks made the same mistakes. Figure 2 shows a confusion matrix of true labels and predicted labels. This figure shows that rather than incorrect relabeling being a random occurrence, the three neural networks were made mistakes between the

same two classes. For example, cats were frequently mistaken for dogs and birds were mistaken for airplanes.

This Noisy Label Correcting Cross Validation had no mechanism to distinguish between hard to classify samples and samples with wrong labels. The same is true with the co-teaching and the INCV methods but they only identify easy and correct samples and ignore not only the wrongly labeled samples but also the hard to classify samples.

5 Future plans

As described above, the deficiency of previous noisy label methods is that they find a subset of the training samples that are labeled correctly *and* easy to classify. This can significantly reduce the size of the training dataset, which reduces the performance of the trained network. To a large extent, these methods trade the reduction in generalization by training with noisy labels with reduction in generalization by training with a smaller training dataset. In this Section we propose novel training approaches that we expect to eliminate the reduction in generalization caused by training with noisy labels.

Based on this new understanding of the noisy label problem, we now propose two new ways to deal with the noisy label problem. One is to use generalization to distinguish between hard, correctly labeled samples and incorrectly labeled samples. The other method is to use soft labels instead of hard labels in order for networks to learn using the full noisy training dataset and decrease the negative generalization impact o the noisy labels.

5.1 Generalization sensitivity analysis

This approach builds on previous noisy label methods, such as co-teaching [5] or INCV [3]. Since those methods extract training samples with small loss, what is left is a combination of hard examples and incorrectly labeled samples. As discussed above, it is difficult to distinguish between these two.

However, there is a way to distinguish between hard examples and incorrectly labeled samples: training on hard examples improves the generalization performance while training on incorrectly labeled samples decreases the generalization performance [14]. In this method we propose to create a highly sensitive test of the generalization performance and use it to determine if a small batch of training data is mostly or all correctly labeled or not.

Algorithm 1 Generalization sensitivity analysis

Require: training dataset D, test dataset T
  Run INCV on $D \rightarrow C, D' = D - C$
  Train network $f(C)$
  Compute $f(T)$
  Choose $k \times numClasses$ hard examples x'
  while $N \neq 0$ **do**
    Split D' into 2 parts
    Separately fine tune f on each part
    Test $f(x')$ on each f
    for Each part **do**
      if $f(x')$ performance increases **then**
        Add this part to C
      else
        Divide this part into two parts
      end if
    end for
  end while

The planned version of the algorithm is shown in Algorithm 1. Specifically, we run INCV or another noisy label method that separates out the easy and correctly labeled samples from the training dataset D into a clean training dataset C. A network $f(C)$ is trained on the clean dataset C to be used through the rest of this algorithm. We assume (and will test) that the definition of a hard example is where the largest outputs of the softmax are numerically close to each other, say within a small value ϵ. For each class we pick k hard test samples (i.e., the largest softmax outputs are within ϵ) where k is a small integer, say between 1 and 5. We expect that this small test set, S, will act as our sensitive barameter for generalization.

The main part of this algorithm is iterative. We divide the training data $D - C$ (i.e., the training data without the clean samples) into two parts. The trained network is fine tuned on each of the two parts and tested on our barameter test set S. If the generalization performance goes up, all or most of the training samples in this part must be clean, hard examples and can be added to C. Otherwise, this part of the data is a mixture of correctly and incorrectly labeled data and should be split into two parts and the procedure should be recursively repeated on the new parts. This loop

is repeated until the size of the training data batch is on the order of a mini-batch.

Of course, there are a number of elements that will become clear as we test this initial method, such as the best stopping point for the iterative procedure, how to change the fine-tuning as the amount of training data decreases, and the magnitude of ϵ. We plan to quickly test our assumptions as we implement this technique.

5.2 Soft labels

The standard way to train neural networks is with 1-hot vectors, which means a label is a vector of zeros with a value of one in the location that corresponds to the class. In other words, the label vector emphatically states that that a training sample is one class and not any of the others. On the other hand, with soft labels, which is related to label smoothing, the label vector can represent probabilities of the training samples being each of the classes. For example, a confusing hard example of a dog, which looks a little like a cat, might be 0.6 in the dog class and 0.4 in the cat class. The confusion matrix in Figure 2 shows there are many examples that fall in this category.

Our initial vision of the algorithm is initiate the training with one hot vectors. Since [1] showed that neural networks learn simple patterns quickly, the loss for the easy and correct samples should go quickly to zero. As training proceeds, the labels for the high loss samples will be incrementally adjusted towards the the predicted classes, perhaps requiring the probability of the original label to be a minimum of 0.5 (this will retain part of the information for hard samples while reducing the effect of false labels).

There are precedents that soft labels will help in the noisy label problem. Mixup [15] is a data augmentation scheme where two training samples from different classes are interpolated to create a mixed sample and a corresponding soft label is used in training. The essence of mixup is to artificially create hard examples and it is one of the best data augmentation methods available. Here, we are not interpolating training samples but rather using the soft labels on naturally occuring hard examples that are part of our training data. We hypothesis that using soft labels on hard examples will increase the generalization performance.

In addition, soft labels will decrease the impact of incorrectly labeled samples because instead of training the network to memorize the wrong label in this instance, soft labels allows for the possibility that this training sample might be a different class. In the wrongly labeled case, it is reasonable to expect that this change in training for wrongly labeled samples will minimize the negative impact on the trained network's generalization.

6 Conclusions

The Noisy Label Correcting Cross Validation method did not improve classification accuracy for a neural network trained on noisy data relative to co-teaching or the INCV methods. As a result, we are not continuing to work with this method.

However, the new understandings learned from this effort inspired two new methods: the generalization sensitivity analysis and soft labels approaches. The generalization sensitivity analysis builds on methods for identifying clean labeled samples by using generalization to distinguish between hard, correctly labeled samples and incorrectly labeled samples in the non-clean remainder. The other method is to incrementally replace hard labels for high loss training samples with soft labels to gain the generalization benefits of the hard training samples and minimize the generalization loss from wrongly labeled samples. We expect this approach to be much more tolerant to wrong labels than the current training methods. Our future plans include testing these methods.

Success of these planned methods will enable the use of automatic labeling methods for new applications, even when a significant fraction of the labels are incorrect. These will lead to a significant reduction in the effort required to obtain large amounts of labeled training data.

References

[1] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 233–242. JMLR. org, 2017.

[2] S. H. Bach, D. Rodriguez, Y. Liu, C. Luo, H. Shao, C. Xia, S. Sen, A. Ratner, B. Hancock, H. Alborzi, et al. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 International Conference on Management of Data*, pages 362–375. ACM, 2019.

[3] P. Chen, B. Liao, G. Chen, and S. Zhang. Understanding and utilizing deep neural networks trained with noisy labels. *arXiv preprint arXiv:1905.05040*, 2019.

[4] J. Goldberger and E. Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

[5] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[6] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017.

[7] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S.-T. Xia, S. Wijewickrema, and J. Bailey. Dimensionality-driven learning with noisy labels. *arXiv preprint arXiv:1806.02612*, 2018.

[8] E. Malach and S. Shalev-Shwartz. Decoupling“ when to update” from“ how to update”. In *Advances in Neural Information Processing Systems*, pages 960–970, 2017.

[9] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[10] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282, 2017.

[11] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[12] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

[13] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[14] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.