TRAINING DEEP NEURAL-NETWORKS BASED ON UNRELIABLE LABELS

Alan Joseph Bekker and Jacob Goldberger

Bar-Ilan University, Israel

ABSTRACT

In this study we address the problem of training a neural network based on data with unreliable labels. We introduce an extra noise layer by assuming that the observed labels were created from the true labels by passing through a noisy channel whose parameters are unknown. We propose a method that simultaneously learns both the neural network parameters and the noise distribution. The proposed method is compared to standard back-propagation neural-network training that ignores the existence of wrong labels. The improved classification performance of the method is illustrated on several standard classification tasks. In particular we show that in some cases our approach can be beneficial even when the labels are set manually and assumed to be error-free.

Index Terms— deep-learning, back-propagation, noisy labels

1. INTRODUCTION

The presence of class label noise inherent to training samples has been reported to deteriorate the performance of even the best classifiers in a broad range of classification problems [1][2]. It was also observed that noisy labels tend to be more harmful than noisy attributes [3]. Finding noisy data is related to the data collection process. Typically, the labels used to train a classifier are assumed to be unambiguous and accurate. However, this assumption often does not hold since the labels provided by human judgments are subjective. Many of the largest image datasets have been extracted from social networks. Because these datasets images are labeled by non-expert users, building a consistent model on a precisely labeled training set is very tedious. Mislabeling examples have been reported even in critical applications such as biomedical datasets where the available data are restricted [4]. A very common approach with noisy datasets is to remove the suspect samples in a preprocessing stage or have them relabeled by a data expert [5]. However, these methods are not scalable and hold the risk of removing crucial examples that may be very significant for small datasets.

Robust noise variants have been proposed for the most common classifiers such as logistic-regression and SVM [6][7]. Natarajan et al. [8] proposed a generic unbiased estimator for binary classification with noisy labels. They developed a surrogate cost function that can be expressed by a weighted sum of the original cost functions, and provided asymptotic bounds for performance. Grandvalet and Bengio [9] addressed the problem of missing labels that can be viewed as an extreme case of noisy label data. They suggested a semi-supervised algorithm that encourages the classifier to predict the non-labeled labels with high confidence by adding a regularization term to the cost function.

In spite of the huge success of deep learning there are not many studies that have explicitly attempted to address the problem of Neural Net (NN) training using data with unreliable labels [10][11][12]. Larsen et al. assumed a single noise parameter that can be calculated by adding a new regularization term and cross validation. Mnih and Hinton [10] proposed a more realistic noise model that depends on the true label. However, they only considered the binary classification case. Sukhbaatar and Fergus [12] recently proposed adding a constrained linear layer at the top of the softmax layer, they have shown that only under some strong assumptions the linear layer can be interpreted as the transition matrix between the true and noisy (observed) labels and the softmax output layer as the true probabilities of the labels.

Unlike previous work, e.g. [13][12][10] in our framework we assume no clean data are available to estimate the noise parameters. We define a probabilistic model for the transformation from true labels to noisy labels and derive a learning scheme based on the EM algorithm. In the E-step we estimate the true labels and in the M-step we apply a back-propagation learning algorithm using the current label estimation. As a by-product of the algorithm we also obtain a parametric description of the noise distribution. The improved results of the proposed approach are demonstrated on several datasets.

2. TRAINING NN WITH NOISY LABELS

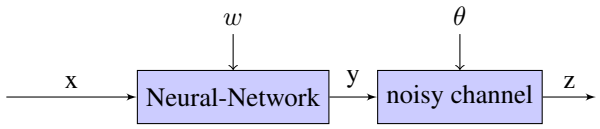
Assume we want to train a multi-class neural-network soft-classifier $p(y = i|x; w)$ where x is the feature vector and w is the network parameter-set. We further assume that in the training process we cannot directly observe the correct label y. Instead, we only have access to a noisy version of it denoted by z. In our approach the noise generation is modeled by a parameter $\theta(i, j) = p(z = j|y = i)$. The noise distribu-

A. J. Bekker is funded in part by the Intel Collaborative Institute for Computational Intelligence (ICRI-CI).

tion is unknown and we want to learn it as part of the training phase. The probability of observing a noisy label z given the feature vector x is:

$$p(z=j|x;w,\theta)=\sum_{i=1}^{k}p(z=j|y=i;\theta)p(y=i|x;w)$$

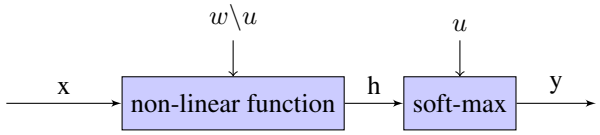
where k is the number of classes. The model is illustrated in the following diagram:



Assume the neural network classifier we are using is based on non-linear intermediate layers followed by a soft-max output layer used for soft classification. Denote the non-linear function applied on an input x by $h=h(x)$ and denote the soft-max layer by:

$$p(y=i|x;w)=\frac{\exp(u_i^\top h)}{\sum_{j=1}^{k}\exp(u_j^\top h)},\qquad i=1,...,k$$

such that $u_1,...,u_k$ are the soft-max parameters which are subset of the entire network parameter set w.



In the training phase we are given n feature vectors $x_1,...,x_n$ with corresponding unreliable labels $z_1,...,z_n$ which are viewed as noisy versions of the correct hidden labels $y_1,...,y_n$. The log-likelihood of the model parameters is:

$$L(w,\theta)=\sum_{t=1}^{n}\log(\sum_{i=1}^{k}p(z_t|y_t=i;\theta)p(y_t=i|x_t;w))$$

Based on the training data, the goal is to find both the noise distribution θ and the Neural Network parameters w that maximize the likelihood function. Since the random variables $y_1,...,y_n$ are hidden, we apply the EM algorithm to find the maximum-likelihood parameter set. The EM-auxiliary function is:

$$Q(w_0,\theta_0,w,\theta)=\sum_{t=1}^{n}\sum_{i=1}^{k}p(y_t=i|x_t,z_t;w_0,\theta_0)\qquad(1)$$

$$\cdot(\log p(y_t=i|x_t;w)+\log p(z_t|y_t=i;\theta))$$

such that w_0,θ_0 are the current parameter values, and we are looking for w,θ that maximize the EM auxiliary function. From the auxiliary function we can easily derive the update procedure of the EM algorithm. In the E-step of each iteration of the EM algorithm we estimate the hidden true data labels based on the noisy labels and the current parameters:

$$\begin{aligned}c_{ti}&=p(y_t=i|x_t,z_t;w_0,\theta_0)\qquad(2)\\&=\frac{p(z_t|y_t=i;\theta_0)p(y_t=i|x_t;w_0)}{\sum_j p(z_t|y_t=j;\theta_0)p(y_t=j|x_t;w_0)}\\&=\frac{\theta_0(i,z_t)\exp(u_{i0}^\top h_0(x_t))}{\sum_j\theta_0(j,z_t)\exp(u_{j0}^\top h_0(x_t))}\end{aligned}$$

such that $u_{10},...,u_{k0}$ and $h_0(x)$ are the components of the current parameter value w_0. In the M-step we update both the NN and the noisy channel parameters. The updated noise distribution has a closed-form solution.

$$\theta(i,j)=\frac{\sum_t c_{ti}1_{\{z_t=j\}}}{\sum_t c_{ti}},\qquad i,j\in\{1,...,k\}\qquad(3)$$

The $k\times k$ matrix θ is a confusion matrix between the soft estimates of the true label $\{c_{ti}|i=1,...,k\}$ and the noisy labels z_t. To find the updated NN parameter w we need to maximize the following function:

$$S(w)=\sum_{t=1}^{n}\sum_{i=1}^{k}c_{ti}\log p(y_t=i|x_t;w)\qquad(4)$$

which is a soft-version of the likelihood function of the fully observed case, based on the current estimate of the true labels.

We next derive the back-propagation equations for the score function (4), which is obtained by the EM algorithm, to get a better understanding of the learning process. In the case that the true labels $y_1,...,y_n$ are given, the partial derivatives of the likelihood function $S(w)=\sum_t\log p(y_t|x_t;w)$ with respect to the soft-max parameters are:

$$\frac{\partial S}{\partial u_i}=\sum_{t=1}^{n}(1_{\{y_t=i\}}-p(y_t=i|x_t;w))h(x_t)\qquad(5)$$

In our case, where only noisy labels $z_1,...,z_n$ are provided, the partial derivatives of the function (4) that we maximize in the M-step are:

$$\frac{\partial S}{\partial u_i}=\sum_{t=1}^{n}(p(y_t=i|x_t,z_t;w_0,\theta_0)-p(y_t=i|x_t;w))h(x_t)\qquad(6)$$

Comparing Eq. (5) and (6) we see that the back-propagation learning algorithm in the case of noisy labels is very similar to the fully observed label case. The only difference is that instead of using the labels values we use estimated labels values that were computed in the E-step based on the current network and noise parameter values. Another consequence of comparing Eq. (5) and (6) is that the computational complexity of the back-propagation algorithm is the same in both cases.

Table 1. The Noisy Labels Neural-Network (NLNN) algorithm.

Input: Data-points $x_1, ..., x_n \in R^d$ with corresponding noisy labels $z_1, ..., z_n \in \{1, ..., k\}$.
Output: Neural-network parameters w and noise parameters θ.

The EM Algorithm iterates between the two steps:

E-step: Estimate true labels based on the current parameter values (2):

$$c_{ti} = p(y_t = i|x_t, z_t; w, \theta)$$

M-step: Update the noise parameter θ:

$$\theta(i,j) = \frac{\sum_t c_{ti} 1_{\{z_t=j\}}}{\sum_t c_{ti}}$$

and train a NN to find w that maximizes the following function:

$$L(w) = \sum_{t=1}^{n} \sum_{i=1}^{k} c_{ti} \log p(y_t = i|x_t; w)$$

There is no need, of course, to fully train the NN model on each EM iteration. Both EM and back-prorogation algorithms are iterative methods and we can alternate between them. For example we can use standard methods for neural-network training and update the noise parameter θ after few passes over the training set. The EM algorithm is a greedy optimization procedure and is notoriously known to be sensitive to the starting point. Hence a good initialization of the model parameters is important to achieve good results. We can use the following strategy to initialize the EM parameters. We first train the NN using standard methods ignoring the fact that the labels are noisy. The obtained NN parameter set w is used as an initial value the EM iteration. We then compute the confusion matrix on the train set and used it as an initial value for the noise parameter set θ:

$$\theta(i,j) = \frac{\sum_t 1_{\{z_t=j\}} p(y_t = i|x_t; w)}{\sum_t p(y_t = i|x_t; w)}, \quad i, j \in \{1, ..., k\}$$

The proposed method, which we dub the Noisy Labels Neural-Network (NLNN) algorithm, is summarized in Table 1.

There have been a number of recent works dealing directly with the issue of training neural-nets based on training data with noisy labels. Reed et al. [14] suggested handling the unreliability of the training data labels by maximizing the likelihood function: with an additional classification entropy regularization term. This cost function, which was studied by Grandvalet and Bengio [9], encourages the model to have a high confidence in predicting labels. This model is advantageous in cases of unlabeled examples because it enables semi-supervised learning. This model, however, do not explicitly address the situation of unreliable data and not provide an explicit modeling of the noisy pattern. The method we present is most closely related to the work of Minh and Hinton [10]. They addressed the problem of mislabeled data points in a particular type of dataset (aerial images). The main difference is that in their approach they assumed that they do not learn the noise parameter. Instead they assume that the noise model can be separately tuned using a validation set or set by hand. This assumption makes the interaction between the EM step and the NN learning much easier since each time a data-point x_t is visited we can compute the $p(y_t = i|x_t, z_t)$ based on the current network parameters and the pre-defined noise parameters.

3. EXPERIMENTS

In this section, we evaluate the robustness of deep learning to training data with noisy labels with and without explicit noise modeling. We used two data-sets with injected label noise in our experiments. The first is the MNIST database of handwritten digits, which consists of 28×28 images. The dataset has 60k images for training and 10k images for testing. The second dataset is used for phoneme classification and is based on the TIMIT acoustic-phonetic continuous speech corpus which has 1.5M frames for training and 500K frames for testing. The input features are the Mel frequency cepstral coefficients (MFCCs) of the signal, powered by delta and delta delta coefficients. Context frames were added to the current time frame as in Mohamed et al. [15]. Overall each speech frame is represented by 351 features. This is the standard feature set for phoneme classification and is known to provide the best classification results.

A similar network architecture and hyper-parameters were used for both datasets. We used a two hidden layer NN comprised of 500 and 300 neurons. The learning scheme is based on a back-propagation algorithm with an adaptive learning rate combined with momentum. The learning rate was initialized to 0.01. It was then increased in each epoch by multiplying the learning rate by 1.05 if the new likelihood exceeds the old likelihood score by more than 4%. Otherwise, the learning rate is kept. If the likelihood score was less than the old likelihood, the learning rate was decreased by multiplying the learning rate by 0.7. The momentum was set to 0.5. The number of training epochs was set to 100 for the naive approach and for the NLNN initialization that was described above. These settings were kept fixed for all the experiments described below.

We generated two types of noisy data from clean data by stochastically changing some of the labels. In the first

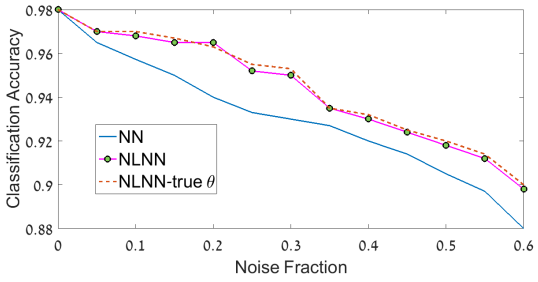


Fig. 1. MNIST test data classification accuracy as a function of fraction of noisy labels with uniform noise.

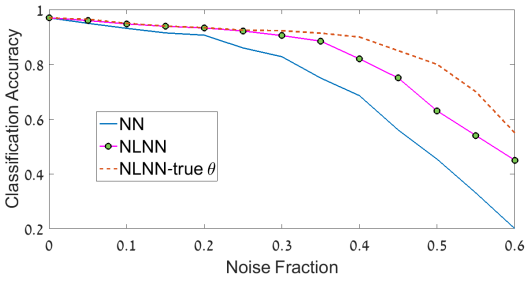


Fig. 2. MNIST test data classification accuracy as a function of fraction of noisy labels with permutation type noise.

type we did not distinguish between classes when intentionally mislabeling data, and rather choosing new labels from a uniform distribution over the labels. The original label i is randomly changed to a different label j with probability $\theta(i,j) = p/(k-1)$ where k is the number of classes and p is the percentage of incorrect labels we want to create in the training data. In the second type we converted each label with probability p to a different label according to a predefined permutation. The labels of the test data remained, of course, unperturbed to validate and compare our method to the regular approach.

MNIST: When training the NLNN model on the MNIST data, we first initialize the network parameters w with a 100 epochs of back propagation using the noisy data and then we run the EM algorithm until the likelihood converges (this took less than 10 iterations). In each iteration we train the NN with the current estimated labels for 50 epochs. In each iteration of the EM we start the back-propagation training with random NN parameters. We found that this strategy works better than starting the NN training with the parameters obtained in the previous EM iteration.

Figure 1 and Figure 2 show comparative test errors results as a function of the fractions of noise for the two noise types described above. We compared the performance of several approaches. We first implemented the naive NN training algorithm (ignoring the existence of noise) and compared it to our model (NLNN) and to a variant of the NLNN model where the

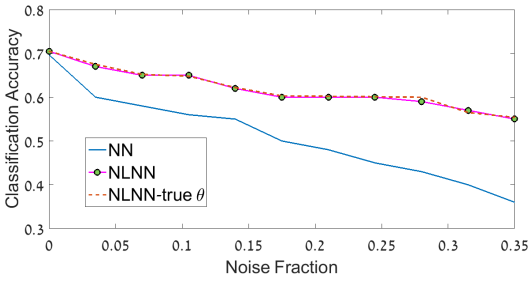


Fig. 3. Phoneme classification accuracy on the TIMIT dataset as a function of fraction of noisy labels.

correct noise distribution is given and is not updated during the EM iterations (NLNN true θ). Figure 1 shows that NLNN significantly outperformed the standard network training approach for every noise ratio and achieved the same results as a model where the known distribution was given. Figure 2 shows the accuracy of the MNIST dataset as a function of the permutation type noise. For a high noise level the NLNN model accuracy dropped compared to the NLNN true θ but still achieved better performance than the naive NN.

TIMIT: In this dataset we conducted an experiment using a uniform noise distribution. Figure 3 shows the phoneme classification accuracy on the test data as a function of the noise level. We show the performance of the naive model, our proposed NLNN model and NLNN model with the true noise distribution θ. Figure 3 shows that the NLNN model achieves significantly better classification accuracy than a naive neural network model that ignores the presence of noisy labels. In this case of phoneme classification we even obtained a slightly better result in the noise-free case where we used the correct phoneme labels provided by the TIMIT dataset. This is an example of a situation where manual annotation is subjective and is not error-free. In the case of speech data, manual phoneme annotation can be noisy because phoneme boundaries are not defined exactly and co-articulation acoustic effects that occur when moving from one phoneme to the next. Classification performance decreased significantly near phoneme boundaries.

To conclude, in this paper we proposed an algorithm for training neural networks based solely on noisy data where the noise distribution is unknown. We showed that we can reliably learn the noise distribution from the noisy data without using any clean data which, in many cases, is not available. The algorithm can be easily incorporated into existing deep learning implementations. Our results encourage collecting more data at a cheaper price, since mistaken data labels can be less harmful to performance. One possible future research direction would be generalizing our learning scheme to cases where both the features and the labels are noisy.

4. REFERENCES

[1] 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," *Artificial intelligence review*, 2010.

[2] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, "Class noise and supervised learning in medical domains: The effect of feature extraction," in *Computer-Based Medical Systems (CBMS)*, 2006.

[3] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, 2004.

[4] U. Alon, n. Barkai, D. Notterman, K. Gish, S.and D. Mack, and A. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[5] C. Brodley and M. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.(JAIR)*, vol. 11, pp. 131–167, 1999.

[6] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[7] B. Jakramate and A. Kabán, "Label-noise robust logistic regression and its applications," in *Machine Learning and Knowledge Discovery in Databases*, pp. 143–158. 2012.

[8] N. Natarajan, I. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[9] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[10] V. Minh and G. Hinton, "Learning to label aerial images from noisy data," in *Int. Conf. on Machine Learning (ICML)*, 2012.

[11] J. Larsen, L. Nonboe, M. Hintz-Madsen, and K. L. Hansen, "Design of robust neural network classifiers," in *Int. Conf. on Acoustics, Speech and Signal Processing*, 1998, pp. 1205–1208.

[12] S. Sukhbaatar and R. Fergus, "Learning from noisy labels with deep neural networks," in *arXiv preprint arXiv:1406.2080*, 2014.

[13] X. Zhu, "Semi-supervised learning literature survey," *Technical Report 1530, University of WisconsinMadison*, 2005.

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

[15] A .Muhamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition.," in *Interspeech*, 2010, pp. 2846–2849.