An Effective Label Noise Model for DNN Text Classification

Ishan Jindal[1], Daniel Pressel[2], Brian Lester[2], and Matthew Nokleby[1]
[1]Wayne State University, Detroit MI 48202
[2]Interactions Digital Roots, Ann Arbor MI 48104
{ishan.jindal, matthew.nokleby}@wayne.edu,
{dpressel, blester}@interactions.com

Abstract

Because large, human-annotated datasets suffer from labeling errors, it is crucial to be able to train deep neural networks in the presence of label noise. While training image classification models with label noise have received much attention, training text classification models have not. In this paper, we propose an approach to training deep networks that is robust to label noise. This approach introduces a non-linear processing layer (*noise model*) that models the statistics of the label noise into a convolutional neural network (CNN) architecture. The noise model and the CNN weights are learned jointly from noisy training data, which prevents the model from overfitting to erroneous labels. Through extensive experiments on several text classification datasets, we show that this approach enables the CNN to learn better sentence representations and is robust even to extreme label noise. We find that proper initialization and regularization of this noise model is critical. Further, by contrast to results focusing on large batch sizes for mitigating label noise for image classification, we find that altering the batch size does not have much effect on classification performance.

1 Introduction

Deep Neural Networks (DNNs) have led to significant advances in the fields of computer vision (He et al., 2016), speech processing (Graves et al., 2013) and natural language processing (Kim, 2014; Young et al., 2018; Devlin et al., 2018). To be effective, supervised DNNs rely on large amounts of carefully labeled training data. However, it is not always realistic to assume that example labels are clean. Humans make mistakes and, depending on the complexity of the task, there may be disagreement even among expert labelers. Further, samples drawn from the class conditional densities with overlapping supports gives rise to the label noise in training datasets. To support noisy labels in data, we need new training methods that can be used to train DNNs directly from the corrupted labels to significantly reduce human labeling efforts. Zhu and Wu (2004) perform an extensive study on the effect of label noise on classification performance of a classifier and find that noise in input features is less important than noise in training labels.

In this work, we add a *noise model* layer on top of our target model to account for label noise in the training set, following (Jindal et al., 2016; Sukhbaatar et al., 2014). We provide extensive experiments on several text classification datasets with artificially injected label noise. We study the effect of two different types of label noise; *Uniform label flipping (Uni)*, where a clean label is swapped with another label sampled uniformly at random; and *Random label flipping (Rand)* where a clean label is swapped with another label from the given number of labels sampled randomly over a unit simplex.

We also study the effect of different initialization, regularization, and batch sizes when training with noisy labels. We observe that proper initialization and regularization helps the noise model learn to be robust to even extreme amounts of noise. Finally, we use low-dimensional projections of the features of the training examples to understand the effectiveness of the noise model.

The rest of the paper is organized as follows. Section 2 discusses the various approaches in literature to handle label noise. In Section 3, we describe the problem statement along with the proposed approach. We describe the experimental setup and datasets in Section 4. We empirically evaluate the performance of the proposed approach along with the discussion in Section 5 and finally conclude our work in Section 6.

Proceedings of NAACL-HLT 2019, pages 3246–3256
Minneapolis, Minnesota, June 2 - June 7, 2019.

2 Related Work

Learning from label noise is a widely studied problem in the classical machine learning setting. Earlier works (Brodley and Friedl, 1999; Rebbapragada and Brodley, 2007; Manwani and Sastry, 2013) consider learning from noisy labels for a wide range of classifiers including SVMs (Natarajan et al., 2013) and fisher discriminants (Lawrence, 2001). Traditional approaches handle label noise by detecting and eliminating the corrupted labels. More details about these approaches can be found in (Frénay and Verleysen, 2014).

Recently, DNNs have made huge gains in performance over traditional methods on large datasets with very clean labels. However large real-world datasets often contain label errors. A number of works have attempted to address this problem of learning from corrupted labels for DNNs. These approaches can be divided into two categories; attempts to mitigate the effect of label noise using auxiliary clean data, and attempts to learn directly from the noisy labels.

Presence of auxiliary clean data: This line of research exploits a small, clean dataset to correct the corrupted labels. For instance, Li et al. (2017) learn a teacher network with clean data to re-weight a noisy label with a soft label in the loss function. Similarly, Veit et al. (2017) use the clean data as a label correction network. One can use this auxiliary source of information to do inference over latent clean labels (Vahdat, 2017). Further, Yao et al. (2018) models the auxiliary trustworthiness of noisy image labels to alleviate the effect of label noise. Though these methods show very promising results, the absence of clean data in some situations might hinder the applicability of these methods.

Learning directly from noisy labels: This research directly learns from the noisy labels by designing a robust loss function, or by modeling the latent labels. For instance, Reed et al. (2014), apply bootstrapping to the loss function to have consistent label prediction for similar images. Similarly, Joulin et al. (2016) alleviate the label noise effect by adequately weighting the loss function using the sample number. Jiang et al. (2017) propose a sequential meta-learning model that takes in a sequence of loss values and outputs the weights for the labels. Ghosh et al. (2017) further explores the conditions on loss functions such that the loss function is noise tolerant.

A number of approaches learn the transition from latent labels to the noisy labels. For example, Mnih and Hinton (2012) propose a noise adaptation framework for symmetric label noise. Based on this work, several other works (Sukhbaatar et al., 2014; Jindal et al., 2016; Patrini et al., 2017; Han et al., 2018) account for the label noise by learning a noisy layer on top of a DNN where the learned transition matrix represents the label flip probabilities. Similarly, Xiao et al. (2015) propose a probabilistic image conditioned noise model. Azadi et al. (2015) proposed an image regularization technique to detect and discard the noisy labeled images. Other approaches include building two parallel classifiers (Misra et al., 2016) where one classifier deals with image recognition and the other classifier models humans reporting bias.

All of these approaches have targeted image classification. In this work, we propose a framework for learning from noisy labels for text classification using a DNN architecture. Similar to (Sukhbaatar et al., 2014; Jindal et al., 2016; Patrini et al., 2017), we append a non-linear processing layer on top of this architecture to model the label noise. This layer helps the base architecture to learn better representations, even in the presence of label noise. We empirically show that, for better classification performance, the knowledge of noise transition matrix is not needed. Instead, the process forces the DNN to learn better sentence representations.

3 Problem Statement

In a supervised text classification setting where $\mathbf{x}_i \in \mathbf{R}^d$ is a d-dimensional word embedding of the ith word in a sentence of length l (padded wherever necessary), we represent the sample as a temporal embedding matrix $\mathbf{X} \in \mathbf{R}^{d \times l}$ which belongs to one of the K classes. Let the noise-free training set be denoted by

$$\mathcal{D} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \cdots, (\mathbf{X}_n, y_n)\},$$

where $y_i \in \{1, \ldots, K\}$ represents the category of ith sample, n is the total number of training samples, and there is an unknown joint distribution $p(\mathbf{X}, y)$ on the sample/label pairs. This temporal representation of a sample $\mathbf{X}$ is fed as input to a classifier on the training set $\mathcal{D}$ with sample categories y. However, as mentioned in Section 2, we cannot access the true noise-free samples labels and instead, observe noisy labels corrupted

by an unknown noise distribution. Let this *noisy* training set be denoted by

$$\mathcal{D}' = \{(\mathbf{X}_1, y_1'), (\mathbf{X}_2, y_2'), \cdots, (\mathbf{X}_n, y_n')\},$$

where y_i' represents the corrupted label for the sentence $\mathbf{X}_i$. In this work, we suppose the label noise is *class-conditional*, where the noisy label y_i' depends only on the true label y_i, but not on the input $\mathbf{X}_i$ or any other labels y_j or y_j'. Under this model, the label noise is characterized by the conditional distribution

$$p(y' = i|y = j) = \Phi_{ij},$$

which we describe via the $K \times K$ column-stochastic matrix Φ_{ij}, parameterized by a matrix $\mathbf{Q} = \{\Phi_{ij}\}$.

In our experiments, we artificially inject label noise into the training and validation sets. We fix the noise distribution Φ_{ij} and, for a training sample, we generate a noisy label by drawing i.i.d from this noise distribution Φ_{ij}. However, we do not alter the test labels.

Though the proposed approach works for any noise distribution, for this study, we focus on two different types of label flip distributions. We use a noise model parameterized by the overall probability of a label error, denoted by $0 \leq p \leq 1$. For a noise level p, we set the noise distribution matrix

$$\Phi = (1-p)\mathbf{I} + \frac{p}{K}\mathbf{I}\mathbf{I}^K, \tag{1}$$

and we call it *Uniform label flip noise model*. Here, $\mathbf{I}$ represents the identity matrix and $\mathbf{I}\mathbf{I}$ denotes the all-ones matrix. Similarly, we describe the *random label flip noise model* as

$$\Phi = (1-p)\mathbf{I} + p\Delta, \tag{2}$$

where $\mathbf{I}$ is the identity matrix, and Δ is a matrix with zeros along the diagonal and remaining entries of each column are drawn uniformly and independently from the $K-1$-dimensional unit simplex. The label error probability for each class is p, while the probability distribution *within* the erroneous classes is drawn uniformly at random.

Our objective is to train a classifier on the noisy labeled sample categories on the training set $\mathcal{D}'$ such that it jointly makes accurate predictions of the true label y and learns the noise transition matrix simultaneously, given $\mathbf{X}$. For the noisy dataset $\mathcal{D}'$, it is straightforward to train a classifier that predicts the *noisy* labels using conditional distribution for the noisy labeled input sentence $\mathbf{X}$:

$$p(y' = \hat{y}'|\mathbf{X}) = \sum_i \Big(p(y' = \hat{y}'|y = \hat{y}_i) \; p(y = \hat{y}_i|\mathbf{X}) \Big). \tag{3}$$

One can learn the classifier associated with $p(y' = \hat{y}'|x)$ via standard training on the noisy set $\mathcal{D}'$. To predict the clean labels by learning the conditional distribution $p(y = \hat{y}_i|x)$ requires more effort, as we cannot extract the "clean" classifier from the noisy classifier when the label noise distribution is unknown.

3.1 Proposed Framework

We refer to the DNN model without the final layer as the *base model* or network *without noise model (WoNM)*. This model, along with the non-linear layer, is trained via back-propagation on the noisy training dataset. The non-linear processing layer in the *noise model* transforms the base model outputs to match the noisy labels during the forward pass better and presents the denoised labels to the base model during the backward pass. The noise layer is parameterized by a square matrix $\Psi \in \mathbf{R}^{K \times K}$). At test time, we remove this learned noise model and use the output of the base model as final predictions.

We refer to the base model parameters as Θ. The base model outputs a probability distribution over the number of K categories denoted as $p\left(y = \hat{y}_i|\mathbf{X}; \Theta\right) \forall i \in \{1, 2, \cdots, K\}$. During the forward pass the noise model transforms this output to obtain the noisy labels as

$$p\left(y'|\mathbf{X}; \Theta, \Psi\right) = \sigma\left(\Psi \times p\left(y|\mathbf{X}; \Theta\right)\right), \tag{4}$$

where $\sigma(\cdot)$ represents the usual softmax operator. Note that both the equations (3) and (4) compute the probability distribution over noisy labels – our noise model does not learn a noise transition matrix. However, we assert that the knowledge of exact noise statistics is neither necessary nor sufficient for the better prediction results.

We learn the base model parameters Θ and the noise model parameters Ψ by maximizing the log likelihood (4) over all of the training samples,

minimizing the *cross-entropy* loss:

$$\mathcal{L}(\Theta, \Psi; \mathcal{D}') = -\frac{1}{n}\sum_{i=1}^{n} \log\left[p\left(y'|\mathbf{X}_i; \Theta, \Psi\right)\right]$$
$$= -\frac{1}{n}\sum_{i=1}^{n} \log\left[\sigma\left(\Psi \times p\left(y|\mathbf{X}_i; \Theta\right)\right)\right]_{y_i} \quad (5)$$

Similar to (Sukhbaatar et al., 2014), we initialize the noise model weights to the identity matrix. Since DNNs have high capacity, we may encounter the situation when the base model itself absorbs all the label noise and, thus, the noise model does not learn anything at all. In order to avoid this situation, and to prevent overfitting, we apply l_2 regularization to the noise model. However, we want the noise model to overfit the label noise. In the experiment section, we observe that with proper regularization and weight initialization the noise model absorbs most of the label noise. Finally, we train the entire network according to the following loss function:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n} \log\left[\sigma\left(\Psi \times p\left(y|\mathbf{X}_i; \Theta\right)\right)\right]_{y_i} + \frac{1}{2}\lambda||\Psi||_2^2. \quad (6)$$

Here, λ is a tuning parameter and we validate the value of λ by repeating the experiment multiple times with multiple λ values over different datasets and choose the one with better classification performance on the validation set for the respective datasets. A value of $\lambda = 0.01$ works best.

4 Datasets and Experimental Setup

In this section, we empirically evaluate the performance of the proposed approach for text classification and compare our results with the other methods.

4.1 General Setting

In all the experiments, we use a publicly-available deep learning library *Baseline* – a fast model development tool for NLP tasks (Pressel et al., 2018). For all the different datasets, we choose a commonly-used, high-performance model from (Kim, 2014) as a base model. To examine the robustness of the proposed approach, we intentionally flip the class labels with 0% to 70% label noise, in other words: $p \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$, and observe the effect of different types of label flipping, such as uniform (*Uni*) and random (*Rand*) label flipping, along with instance-dependent label noise. For all the experiments, we use early stopping based on validation set accuracy where the class labels in validation are also corrupted.

	Dataset	K	L	N	T	Type
Text Data	SST-2	2	19	76961	1821	Balanced
	TREC	6	10	5000	500	Not Balanced
	AG-News	4	38	110K	10K	Balanced
	DBpedia	14	29	504K	70K	Balanced

Table 1: Summary of text classification datasets; K: denotes the number of classes, L: represents the average length of sentence, N: denotes the number of training samples, T: represents the number of test samples, Type: describes whether the dataset is balanced.

We indicate the performance of a standard deep network *Without Noise model (WoNM)* on the noisy label dataset. We also plot the results for the stacked *Noise Model Without Regularization (NMWoRegu)* and stacked *Noise Model With Regularization (NMwRegu)*. Unless otherwise stated, in all the deep networks with the stacked noise model, we initialize the noise layer parameters as an identity matrix. We further analyze the effect of the noise layer initialization on the overall performance. We define *TDwRegu* as the stacked noise model with regularization, initialized with true injected noise distribution and *RandwRegu* as the stacked noise model with regularization, initialized randomly. We run all experiments five times and report the mean accuracy.

4.2 Datasets

Here, we describe all the text classification datasets used to evaluate the performance of the proposed approach. The base model architecture is the same for all datasets. For each set, we tune the number of filter windows and filter lengths using the development set. Along with the description, we also provide the hyper-parameters we selected for each. Table 1 summarizes the basic statistic of the datasets.

1. SST-2[1] (Socher et al., 2011): Stanford Sentiment Treebank dataset for predicting the sentiment of movie reviews. The classification task involves detecting positive or negative reviews. Using the base model with

[1] http://nlp.stanford.edu/sentiment/

		Batch Size	50							100							
		Label Flips	Random							Random							
SST2	Noise%	Clean Labels	10	20	30	40	45	47	50	0	10	20	30	40	45	47	50
	WoNM	87.27%	83.29%	79.08%	73.42%	64.03%	58.1%	54.73%	49.7%	86.53%	81.44%	75.58%	71.88%	63.39%	57.12%	55.81%	52.32%
	TDwRegu01	86.88%	85.37%	84.92%	**83.29%**	**78.53%**	**74.01%**	51.95%	**49.5%**	**86.88%**	**84.88%**	**85.08%**	**82.41%**	**76.09%**	**70.10%**	**58.98%**	49.86%
	NMWoRegu	**87.28%**	**86.2%**	84.07%	81.29%	70.42%	62.27%	**55.76%**	48.42%	86.66%	84.72%	83.03%	78.2%	66.65%	61.32%	57.11%	**52.24%**
	NMwRegu001	86.08%	85.01%	83.82%	81.97%	73.18%	62.18%	55.63%	48.87%	**86.51%**	**85.26%**	84.37%	81.05%	69.54%	60.89%	56.8%	51.6%
	NMwRegu01	**87.78%**	**86.04%**	**85.04%**	**82.7%**	**77.43%**	**66.96%**	**61.5%**	**49.08%**	86.33%	85.17%	**85.10%**	**81.9%**	**76.2%**	**65.47%**	**58.92%**	**52.46%**
		Batch Size	10														
		Label Flips	Uniform							Random							
TREC	Noise%	Clean data	10	20	30	40	50	60	70	0	10	20	30	40	50	60	70
	WoNM	92.8%	87.6%	83.6%	75.87%	67.27%	57.4%	46.27%	42.8%	92.8%	85.93%	82.2%	74.0%	68.4%	53.53%	48.2%	31.47%
	TDwRegu01	50.87%	45.33%	45.4%	36.33%	25.87%	28.33%	16.87%	16.87%	50.87%	56.4%	36.8%	24.0%	25.47%	22.6%	18.8%	22.6%
	NMWoRegu	92.33%	88.07%	84.67%	76.4%	68.47%	58.4%	50.07%	41.33%	92.07%	85.87%	84.27%	72.47%	66.53%	50.13%	44.6%	33.0%
	NMwRegu001	92.47%	90.53%	88.07%	81.6%	73.47%	64.07%	55.87%	43.67%	92.4%	88.53%	86.4%	77.2%	67.67%	54.67%	47.93%	**34.87%**
	NMwRegu01	**92.73%**	**90.8%**	**89.53%**	**88.67%**	**84.93%**	**79.67%**	**69.67%**	**52.4%**	**92.7%**	**90.33%**	**90.6%**	**86.47%**	**83.07%**	**70.93%**	**65.2%**	33.4%
		Batch Size	50														
		Label Flips	Uniform							Random							
	Noise%	Clean Labels	10	20	30	40	50	60	70	0	10	20	30	40	50	60	70
	WoNM	92.8%	87.27%	83.07%	75.00%	69.13%	61.53%	50.13%	39.8%	92.8%	86.00%	81.2%	76.2%	64.07%	52.4%	47.4%	34.13%
	TDwRegu01	55.73%	50.4%	44.73%	39.6%	22.27%	25.67%	14.93%	21.00%	55.73%	45%	44.93%	27.73%	27.87%	22.6%	17.87%	22.6%
	NMWoRegu	92.6%	87.73%	83.33%	76.33%	70.67%	56.8%	48.2%	39.67%	92.60%	85.27%	83.00%	73.6%	65.8%	50.4%	45.93%	30.73%
	NMwRegu001	92.53%	90.73%	87.20%	82.53%	73.93%	65.07%	52.87%	44.60%	92.53%	88.%	87.2%	79.07%	71.2%	51.67%	49.00%	33.40%
	NMwRegu01	**92.53%**	**91.33%**	**90.27%**	**88.47%**	**83.87%**	**77.87%**	**68.73%**	**55.67%**	**92.53%**	**90.00%**	**90.2%**	**85.93%**	**82.6%**	**71.4%**	**67.33%**	**37.53%**
		Batch Size	100														
		Label Flips	Uniform							Random							
AG-News	Noise%	Clean Labels	10	20	30	40	50	60	70	0	10	20	30	40	50	60	70
	WoNM	92.31%	89.96%	87.42%	84.55%	79.96%	75.42%	68.78%	59.94%	92.31%	89.71%	86.11%	79.05%	76.04%	65.09%	45.79%	38.12%
	TDwRegu01	92.47%	92.25%	92.15%	92.04%	84.87%	77.56%	63.13%	47.83%	92.68%	92.09%	91.99%	61.81%	62.44%	70.26%	24.99%	38.12%
	NMWoRegu	91.94%	91.89%	91.21%	90.51%	89.29%	88.02%	86.25%	79.88%	91.97%	91.79%	91.00%	90.04%	88.82%	86.49%	**77.66%**	43.01%
	NMwRegu001	92.47%	92.21%	91.82%	91.21%	90.71%	89.61%	88.43%	85.32%	**92.62%**	92.14%	91.5%	91.07%	90.2%	88.68%	64.01%	55.11%
	NMwRegu01	**92.55%**	**92.23%**	**92.2%**	**91.98%**	**91.7%**	**91.23%**	**90.54%**	**89.78%**	92.57%	**92.23%**	**91.96%**	**91.69%**	**91.13%**	**90.77%**	76.64%	**62.04%**
		Batch Size	1024														
		Label Flips	Uniform							Random							
	Noise%	Clean Labels	10	20	30	40	50	60	70	0	10	20	30	40	50	60	70
	WoNM	92.42%	89.77%	87.04%	84.07%	79.77%	74.54%	67.59%	59.41%	92.29%	89.47%	85.78%	80.51%	75.99%	65.55%	45.50%	39.75%
	TDwRegu01	92.61%	92.37%	92.18%	92.07%	84.92%	62.74%	63.43%	47.59%	92.54%	92.34%	91.82%	53.81%	69.04%	48.88%	25.05%	46.9%
	NMWoRegu	92.16%	91.51%	90.80%	89.58%	85.58%	79.96%	70.79%	62.89%	92.22%	91.61%	90.33%	86.92%	82.61%	71.49%	48.96%	39.96%
	NMwRegu001	92.4%	92.13%	91.88%	91.46%	90.14%	89.07%	86.96%	80.94%	92.54%	91.87%	91.38%	90.42%	99.18%	86.78%	75.74%	50.11%
	NMwRegu01	**92.66%**	**92.2%**	**92.29%**	**92.09%**	**91.7%**	**91.24%**	**90.72%**	**89.88%**	**92.57%**	**92.11%**	**91.99%**	**91.57%**	**91.2%**	**90.5%**	**77.93%**	**61.12%**
		Batch Size	512														
		Label Flips	Uniform							Random							
DBpedia	Noise%	Clean Labels	30	50	70	75	80	85	90	0	30	50	70	75	80	85	90
	WoNM	99.01%	95.19%	89.59%	74.01%	67.73%	57.87%	47.48%	34.01%	99.01%	94.72%	86.08%	62.87%	53.13%	40.78%	26.6%	12.42%
	NMWoRegu	98.93%	95.07%	90.2%	78.32%	73.65%	66.24%	54.24%	40.9%	98.93%	93.55%	84.53%	25.96%	54.84%	42.96%	29.25%	12.97%
	NMwRegu001	**99.04%**	**98.94%**	**98.81%**	**98.61%**	**98.52%**	**98.33%**	**98.13%**	**97.53%**	**99.04%**	98.93%	98.82%	98.62%	**98.48%**	**98.33%**	**89.00%**	11.36%
	NMwRegu01	99.01%	98.89%	98.71%	98.45%	98.32%	98.10%	97.76%	97.15%	98.92%	**99.01%**	**98.88%**	**98.72%**	98.10%	97.67%	38.62%	**16.27%**
		Batch Size	1024														
		Label Flips	Uniform							Random							
	Noise%	Clean Labels	30	50	70	75	80	85	90	0	30	50	70	75	80	85	90
	WoNM	98.96%	97.93%	96.47%	90.49%	68.07%	59.78%	48.06%	55.29%	98.96%	94.75%	86.36%	63.75%	53.39%	40.87%	26.18%	11.9%
	NMWoRegu	98.87%	97.37%	95.71%	89.54%	72.79%	66.49%	55.27	60.7%	98.87%	93.96%	85.6%	44.85%	54.32%	42.21%	28.63%	12.51%
	NMwRegu001	**98.97%**	**98.9%**	**98.79%**	**98.53%**	**98.50%**	**98.32%**	**98.19%**	**97.27%**	**98.97%**	98.83%	98.51%	98.1%	**98.49%**	**98.32%**	**83.79%**	10.51%
	NMwRegu01	98.92%	98.79%	98.58%	98.26%	98.32%	98.09%	97.79%	96.54%	98.92%	**98.88%**	**98.72%**	**98.35%**	98.12%	97.72%	33.10%	**15.94%**

Table 2: Test performance for different text classification datasets

clean labels we obtain classification accuracy of 87.27%. For this dataset, the base model network architecture consists of an input and embedding layer + $[3, 4, 5]$ feature windows with 100 feature maps each and dropout rate 0.5 with batch size 50.

2. TREC[2] (Voorhees and Tice, 1999): A question classification dataset consisting of fact based questions divided into broad semantic categories. We use a six-class version of TREC dataset. For this dataset, the base model network architecture consists of an input and embedding layer + $[3]$ one feature windows with 100 feature maps and dropout rate 0.5 with batch size 10.

3. Ag-News[3] (Zhang et al., 2015): A large-scale, four-class topic classification dataset. It contains approx 110K training samples. For this dataset, the base model network architecture consists of Input layer + Embedding layer + $[3, 4, 5]$ feature windows with 200 feature maps and dropout rate 0.5 with batch size 100.

4. DBpedia[3] (Zhang et al., 2015): A large scale 14-class topic classification dataset containing $36K$ training samples per category. For this dataset, the base model network architecture consists of Input layer + Embedding layer + $[1, 2, 3, 4, 5, 7]$ feature windows with 400 feature maps each and dropout rate 0.5 with batch size 1024.

[2] http://cogcomp.cs.illinois.edu/Data/QA/QC/

[3] http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

For all the datasets, we use Rectified Linear Units (ReLU) and fix the base model architecture. We use early stopping on dev. sets for all the datasets. We run all the experiments 5 times and report the average classification accuracy in Table 2. We train all the networks end-to-end via stochastic gradient descent over shuffled mini-batches with the Adadelta update rule (Zeiler, 2012) except for the DBpedia, where we use SGD. In order to improve base model performance, we initialize the word embedding layer with the publicly available *word2vec* word vectors (Mikolov et al., 2013) for all the datasets except for DBpedia, where we use *GloVe* embeddings (Pennington et al., 2014).

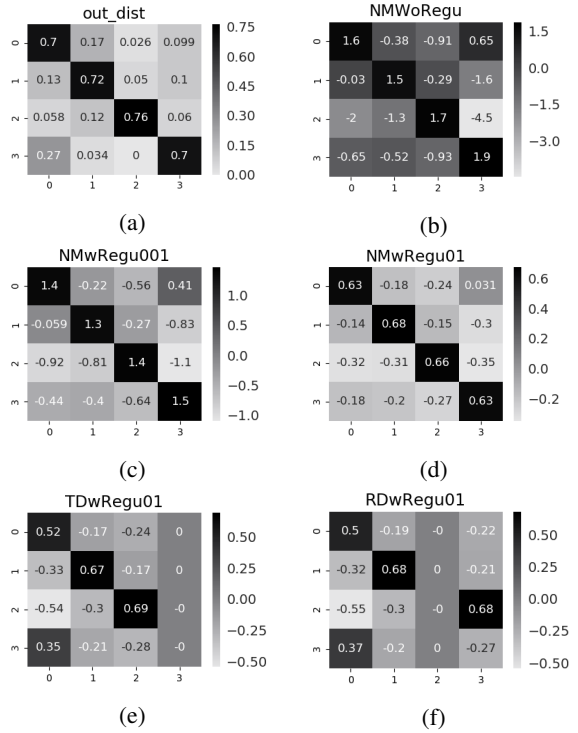


Figure 1: AG-News Dataset: a) Input random label noise; (b-f) learned weight matrix learned by different noise models.

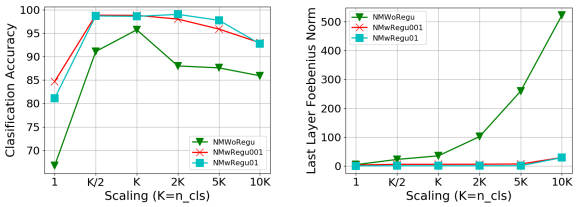
5 Results and Discussion

We evaluate the performance of our model in Table 2 for each datasets in the presence of uniform and random label noise and compare the performance with the base model (*WoNM*) as our baseline. For all datasets, the proposed approach is significantly better than the baseline for both random and uniform label noise. For all datasets, we observe a gain of approximately 30% w.r.t the baseline in the presence of extreme label noise. We do observe a drop in classification accuracy as we increase the percentage of label noise but even at the extreme label noise our method outperformed the baseline method. Interestingly, if we assume an oracle to determine prior knowledge of true noise distribution (*TDwRegu01*), it does not necessarily improve classification performance, especially for multi-class classification problems. For binary classification, using the SST-2 dataset, we did observe that the noise model initialized with the true noise distribution works better than all the other models. In addition to this, we also observe a slight performance gain for the proposed approach over the baseline with clean labels – perhaps due to label noise inherent in the datasets.

5.1 Effect of Different Regularizers

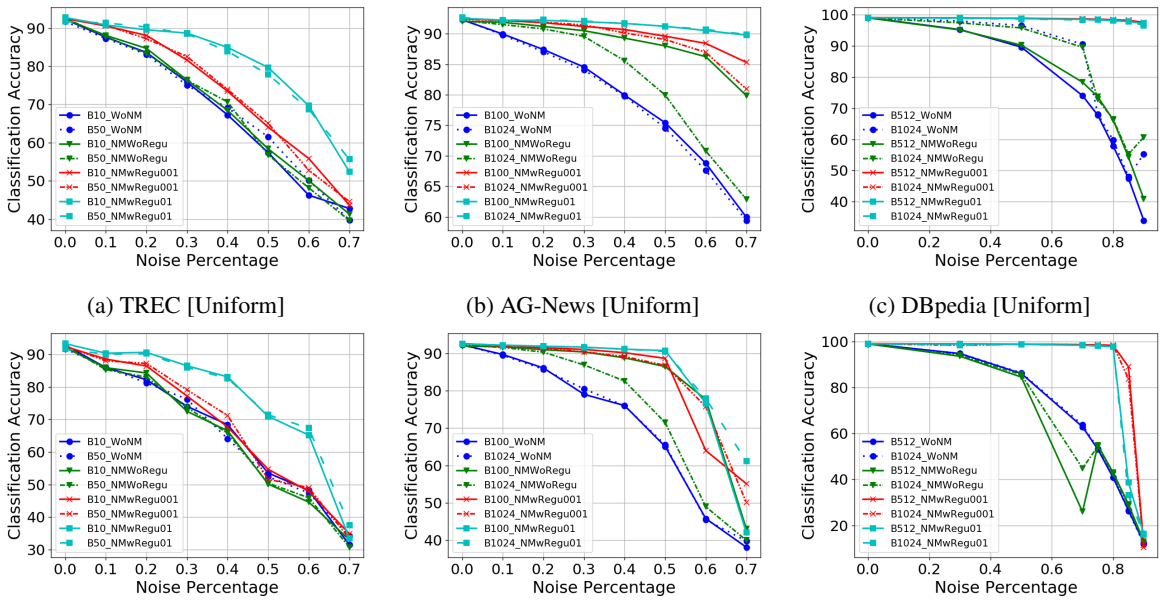
The *NMwRegu01* performs better in all cases for both types of label noise. We plot the weight matrix learned by all the noise models in all the noise regimes. For brevity, we only plot the weight matrix for AG-News datasets with 30% label noise in Fig.1. We find that l_2 regularization diffuses the diagonal weight elements and learns more smoothed off-diagonal elements which resemble the corresponding input label noise distribution in Fig. 1d. This also means that, without regularization, the noise model has less ability to diffuse the diagonal elements which leads to poor classification performance. Therefore, we use a regularizer (l_2) to diffuse the diagonal entries.

In some cases, especially for low label noise, we find the l_2 regularization with a small penalty works better than a large penalty since, for low label noise, learning a less diffuse noise is beneficial. The proposed approach scales to a large number of label categories, as evident from the experiments on DBpedia dataset in the last row of Table .2.



(a) Classification accuracy (b) Noise model norm

Figure 2: Effect of noise model initialization scaling on the classification performance



(a) TREC [Uniform] (b) AG-News [Uniform] (c) DBpedia [Uniform]

(d) TREC [Random] (e) AG-News [Random] (f) DBpedia [Random]

Figure 3: Effect of batch size on label noise classification for different datasets. [Best viewed in color]

5.2 Effect of Different Scaling Factors on Noise Layer Initialization

We initialize the noise model weights as identity matrices with gain equal to the number of classes (gain = K) for all experiments. We observe the effect of different gain values on the overall performance of the proposed network in Fig. 2 where on x-axis we vary the scaling factor – a function of number of classes in the dataset. We plot the classification performance for the DBpedia dataset with 50% random noise. For each noise model in Fig. 2a, we find that setting the gain to K works best and any other gain results in poor performance.

In Fig. 2b we plot the Frobenius norm of the learned noise model weights with respect to the different gain values. We find that, using the high gain initialization, the model learns a high noise model norm, resulting in poor classification performance. This finding provides support to the claim in (Liao et al., 2018) that "higher layer norm leads to highest test errors."

5.3 Effect of Batch Size

We also observe the effect of different batch sizes on performance as described in (Rolnick et al., 2017). For all datasets, we do observe small performance gains for highly non-uniform noisy labels, for instance 70%, in Fig. 3 row 2. However, for uniform label flips, we do not observe performance gains with increasing batch size.

	TRB			TRPr		
Data(N%)	WoNM	Noisy	True	NMwRegu01	Noisy	True
SST2 (40%)	70.24	70.95	79.24	82.32	73.90	83.25
AG (70%)	59.70	52.44	79.18	90.33	86.27	89.4
AG (60%)	83.25	68.8	88.28	90.45	87.77	90.78
TREC (40%)	66.80	63.4	79.0	73.40	69.6	83.2
TREC (20%)	83.6	80.0	86.0	87.40	83.6	90.0

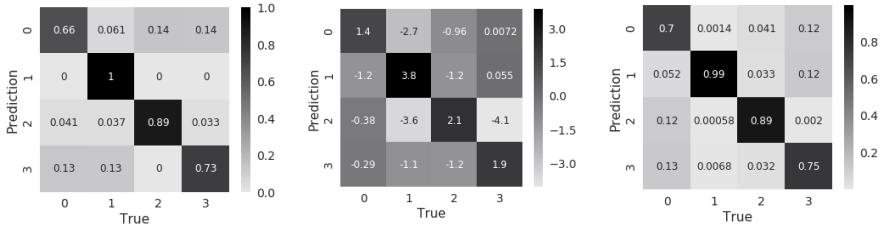
Table 3: SVM Classification

5.4 Instance Dependent Label Noise

We further investigate the performance of the proposed approach on instance-dependent label noise by flipping each class labels with different noise percentages as shown in Fig. 4a. For brevity, we present results on AG-News dataset in Fig. 4. On this type of label noise, the performance of the proposed approach is far better than the baseline with a performance improvement of $\sim 6\%$. The noise model learned by the proposed approach is shown in Fig. 4b and we show the normalized weight matrix in Fig. 4c. We observe that the learned noise model is able to capture the input label noise statistics and is highly correlated to the input noise distribution with Pearson Correlation Coefficient 0.988.

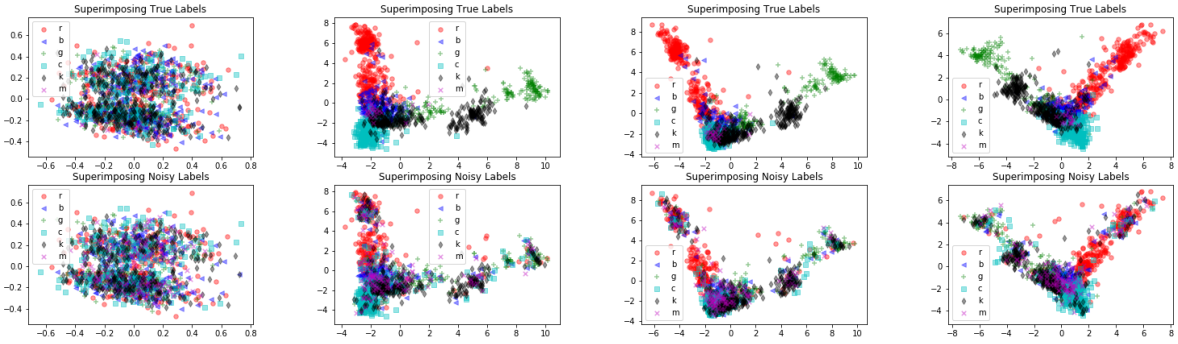
5.5 Understanding Noise Model

In order to further understand the noise model, we first train the base model and the proposed model on noisy labels. Afterward, we collect



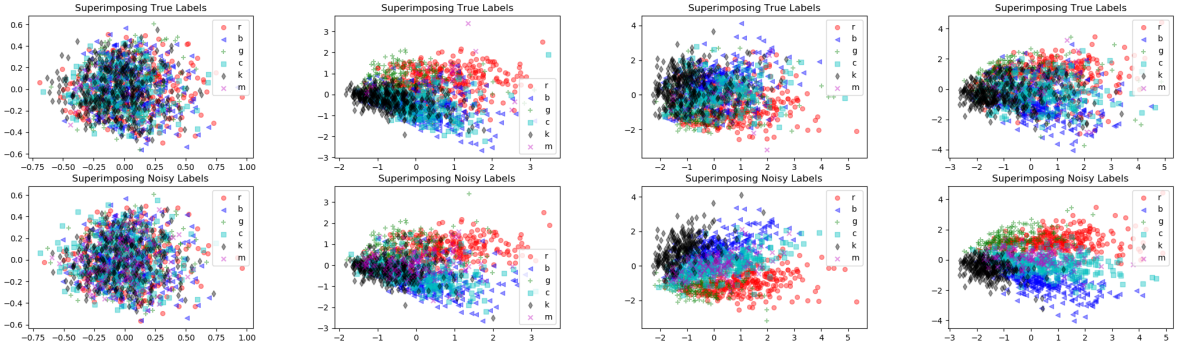
(a) (b) (c)

Figure 4: AG-News Dataset: a) input instance dependent label noise; b) learned weight matrix by proposed approach; c) column normalization of (b).



Iteration 0 Iteration 5 Iteration 10 Iteration 18

(a) Proposed model



Iteration 0 Iteration 5 Iteration 10 Iteration 18

(b) No noise model stacked

Figure 5: t-SNE visualization of the last layer activations of a base network before softmax for TREC Dataset with 50% corrupted labels; First row in (a) when the corresponding true labels are superimposed on the t-SNE data points; Second row in (a) when the noisy labels are superimposed onto the t-SNE data points. [Best viewed in color]

the last fully-connected layer's activations for all the training samples and treat them as the learned feature representation of the input sentence. We get two different sets of feature representations, one corresponding to the base model (*TRB*), and the other corresponding to the proposed model (*TRPr*). Given these learned feature representations – the artificially injected noisy labels and the true labels of the training data – we learn two different SVMs for each model, with and without noise. For the base model, for both SVMs, we use TRB representation as inputs and train the first SVM with the true labels as targets and the second SVM with the unreliable labels as targets. Simi-

larly, we train two SVMs for the proposed model. After training, we evaluate the performance of all the learned SVMs on clean test data in Table 3, where the 1st column represents the corresponding model performance, "Noisy" and "True" column represents the SVM performance when trained on noisy and clean labels, respectively. We run these experiments for different datasets with different label noise.

The SVM, trained on TRB and noisy labels, is very close to the base model performance (3). This suggests that the base model is just fitting the noisy labels. On the other hand, when we train an SVM on the TRPr representations with true labels as targets, the SVM achieves the proposed model performance. This means that the proposed approach helps the base model to learn better feature representations even with the noisy targets, which suggest that this noise model is learning a label denoising operator.

We analyze the representation of training samples in feature domain by plotting the t-SNE embeddings (Van Der Maaten, 2014) of the TRB and TRPr. For brevity, we plot the t-SNE visualizations for TREC dataset with 50% label noise in Fig. 5 .

For each network, we show two different t-SNE plots. For example in Fig. 5a we plot two rows of t-SNE embeddings for the proposed model. In the first row of Fig. 5a, each training sample is represented by its corresponding true label, while in the second row (the noisy label plot) each training sample is represented by its corresponding noisy label. We observe that, as the learning process progresses, the noise model helps the base model to cluster the training samples in the feature domain. With each iteration, we can see the formation of clusters in Row 1. However, in Row 2, when the noisy labels are superimposed, the clusters are not well separated. This means that the noise model denoises the labels and presents the true labels to the base network to learn.

In Fig. 5b, we plot two rows of t-SNE embeddings of the TRB representations. It seems that the network directly learns the noisy labels. This provides further evidence to support (Zhang et al., 2016)'s finding that the deep network memorizes data without knowing of true labels. In Row 2 of Fig. 5b, we can observe that the network learns noisy features representations which can be well clustered according to given noisy labels.

6 Conclusion

In this work, we propose a framework to enable a DNN to learn better sentence representations in the presence of label noise for text classification tasks. To model the label noise, we append a non-linear noise model on top of the standard DNN architecture. With proper initialization and regularization, the noise model is able to absorb most of the label noise and helps the base model to learn better sentence representations.

Acknowledgments

We thank the anonymous reviewers for their detailed and insightful comments. We would also like to thank Patrick Haffner, Sagnik Ray Choudhury, Yanjie Zhao and Amy Hemmeter for their valuable discussions with us during the course of this research.

References

Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. 2015. Auxiliary image regularization for deep cnns with noisy labels. *arXiv preprint arXiv:1511.07069*.

Carla E Brodley and Mark A Friedl. 1999. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Benoît Frénay and Michel Verleysen. 2014. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *AAAI*, pages 1919–1925.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. 2018. Masking: A new perspective of noisy supervision. *arXiv preprint arXiv:1805.08193*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2017. Mentornet: Regularizing very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. 2016. Learning deep networks from noisy labels with dropout regularization. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 967–972. IEEE.

Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. 2016. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Neil D Lawrence. 2001. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML*. Citeseer.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936.

Qianli Liao, Brando Miranda, Andrzej Banburski, Jack Hidary, and Tomaso Poggio. 2018. A surprising linear relationship predicts test performance in deep networks. *arXiv preprint arXiv:1807.09659*.

Naresh Manwani and PS Sastry. 2013. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. 2016. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2939.

Volodymyr Mnih and Geoffrey E Hinton. 2012. Learning to label aerial images from noisy data. In *Proceedings of the 29th International conference on machine learning (ICML-12)*, pages 567–574.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. 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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Daniel Pressel, Sagnik Ray Choudhury, Brian Lester, Yanjie Zhao, and Matt Barta. 2018. Baseline: A library for rapid modeling, experimentation and development of deep learning algorithms targeting nlp. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 34–40. Association for Computational Linguistics.

Umaa Rebbapragada and Carla E Brodley. 2007. Class noise mitigation through instance weighting. In *European Conference on Machine Learning*, pages 708–715. Springer.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Arash Vahdat. 2017. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, pages 5596–5605.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 6575–6583.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699.

Jiangchao Yao, Jiajie Wang, Ivor W Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang. 2018. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Xingquan Zhu and Xindong Wu. 2004. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210.