Making Convex Loss Functions Robust to Outliers using $e$-Exponentiated Transformation

Suvadeep Hajra

Abstract

In this paper, we propose a novel $e$-exponentiated transformation, $0.5 < e < 1$, for loss functions. When the transformation is applied to a convex loss function, the transformed loss function enjoys the following desirable property: for one layer network, all the stationary points of the empirical risk are global minima. Moreover, for a deep linear network if the risk is differentiable at all the local minima and the hidden layers are at least as wide as either the input layer or the output layer, all local minima can be shown to be a global minimum. Using a novel generalization error bound, we have theoretically shown that the transformed loss function has a tighter bound for datasets corrupted by outliers. Our empirical observation shows that the accuracy obtained using the transformed loss function can be significantly better than the same obtained using the original loss function and comparable to that obtained by some other state of the art methods in the presence of label noise.

1. Introduction

Convex loss functions are widely used in machine learning as their usage lead to convex optimization problem in a single layer neural network or in a kernel method. That, in turn, provides the theoretical guarantee of getting a globally optimum solution efficiently. However, many earlier studies have pointed out that convex loss functions are not robust to outliers (Long & Servedio, 2008; 2010; Ding & Vishwanathan, 2010; Denchev et al., 2012; Manwani & Sastry, 2013; Ghosh et al., 2015). Though, many nonconvex loss functions provide some theoretical guarantees with respect to robustness to the label noise, they suffer from either poor local minima or larger computational cost (Ding & Vishwanathan, 2010; Rooyen et al., 2015).

In this paper, we try to bridge the gap between the usage of convex loss functions and robustness to outliers by proposing $e$-exponentiated transformation of convex loss functions. Given a convex loss function $l(\hat{y}, y)$, we define it’s $e$-exponentiated transformation to be $l^{e,c}(\hat{y}, y) = l(\sigma^{e,c}(\hat{y}), y)$ for $0.5 < e < 1$ and some real positive constant $c$ where $\sigma^{e,c}(\hat{y})$ is given by

$$
\sigma^{e,c}(\hat{y}) = \begin{cases} 
\text{sgn}(\hat{y})|\hat{y}|^e & \text{if } |\hat{y}| \geq c \\
\text{otherwise} & \text{otherwise}
\end{cases}
$$

(1)

with $|\hat{y}|$ denoting the absolute value of $\hat{y} \in \mathbb{R}$ and the sign function $\text{sgn}(\hat{y})$ defined to be equal to 1 for $\hat{y} \geq 0$, -1 otherwise. For a differentiable convex loss function $l(\cdot, \cdot)$, its $e$-exponentiated transformation $l^{e,c}(\cdot, \cdot)$ possesses the following properties:

1. It is differentiable everywhere except at $\hat{y} \in \{-c, c\}$.
2. It is nonconvex.
3. All the stationary points of the empirical risk of a single layer neural network with $e$-exponentiated loss function are also global minima.
4. If the gradient exists at all the local minima and the hidden layers are sufficiently wide, all the local optima of the empirical risk of a deep linear network with $e$-exponentiated loss are also global minima.

Note that the first property ensures that a gradient based optimization algorithm can be used for empirical risk minimization with $e$-exponentiated loss function. The second property helps to overcome the limitation of convex loss function in the presence of outliers. Indeed, an $e$-exponentiated loss function $l(\hat{y}, y)$ is more robust to outliers than the corresponding convex loss function $l(\hat{y}, y)$ as the slope $\left| \frac{d}{d\hat{y}} l^{e,c}(\hat{y}, y) \right| = e|\hat{y}|^{e-1} \left| \frac{d}{d\sigma^{e,c}(\hat{y})} l(\sigma^{e,c}(\hat{y}), y) \right| < \left| \frac{d}{d\hat{y}} l(\hat{y}, y) \right|$
Additionally, by introducing a novel generalization error bound (Rosasco et al., 2004) which strongly depends on the Lipschitz constant of a loss function, our derived bound depends on the Lipschitz constant only weakly. Consequently, even though the loss function is nonconvex, the bound can be tighter for a loss function compared to the corresponding convex loss. Using the bound, we have also shown that the bound can be tighter for $e$-exponentiated loss function than that obtained by the corresponding convex loss function. In Section 4, we have shown our experimental result. Finally, Section 5 concludes the work.

2. Empirical Risk Minimization Using $e$-Exponentiated Loss

Without loss of generality, we consider the empirical risk minimization of some single layer neural network with $e$-exponentiated loss function for a binary classification problem. Given a convex loss function $l(\cdot, \cdot)$, the empirical risk minimization of a single layer neural network is given by:

$$\hat{R}_t(w; D) = \frac{1}{N} \sum_{i=1}^{N} l(\hat{y}_i, y_i) = \frac{1}{N} \sum_{i=1}^{N} l(w^T \phi(x_i), y_i)$$

where $D = \{(x_i, y_i)\}_{i=1}^{N}$ is the training set, $\phi(x) \in \mathbb{R}^d$ is the feature representation of the sample $x$ and the target $y_i$s takes a value from $\{1, -1\}$ for $i \in \{1, \cdots, N\}$. The corresponding empirical risk with $e$-exponentiated loss $l_{e,c}(\cdot, \cdot)$ is given by:

$$\hat{R}_{t,e,c}(w; D) = \frac{1}{N} \sum_{i=1}^{N} l_{e,c}(\hat{y}_i, y_i) = \frac{1}{N} \sum_{i=1}^{N} l(\sigma^{e,c}(\hat{y}_i), y_i)$$

where $e \in (0.5, 1)$, $c > 0$ and $\sigma^{e,c}(\cdot)$ is as defined in Eq. (1). In the rest of the paper, we will ignore the second argument of $\hat{R}_t(w; D)$ and $\hat{R}_{t,e,c}(w; D)$ whenever $D$ can be inferred from the context.
In the next section, we show that all the stationary points of \( \hat{R}_{\epsilon,c}(w) \) are also global optima.

### 2.1. All Stationary Points are Global Minima

As mentioned before, \( \hat{R}_{\epsilon,c}(w) \) is no more a convex function with respect to the parameter \( w \). Now we show that even if the empirical risk \( \hat{R}_{\epsilon,c}(w) \) is no more convex, all of its stationary points are global minima. Towards that aim, we prove the following lemma which states a relaxed version of the first order convexity condition for \( \hat{R}_{\epsilon,c}(w) \).

**Lemma 1** Let \( w \in B \subset \mathbb{R}^d \) where \( B \) is a closed ball of radius \( \delta \) with origin as the center. Let us denote \( \hat{R}_{\epsilon,c}(w) \) by \( f(w) \). Then for any two points \( w_1, w_2 \in B \) such that \( f(w) \) is differentiable at \( w_1 \), the following inequality holds:

\[
\langle -\nabla_w f(w_1), w_2 - w_1 \rangle \geq \frac{1}{2} \epsilon (f(w_1) - f(w_2))
\]

where \( \nabla_w f(w_1) \) denotes the derivative of \( f \) with respect to \( w \) at \( w = w_1 \) and \( \langle \cdot, \cdot \rangle \) denotes the inner product between two vectors in \( \mathbb{R}^d \).

Proof has been given in Appendix A. In the next theorem, we state our main result of this section.

**Theorem 1** Let all the conditions of Lemma 1 are satisfied. Then all the stationary points of \( f(w) = \hat{R}_{\epsilon,c}(w) \) are global minima.

**Proof**: Let \( w_0 \in B \) is a point satisfying \( \nabla_w f(w_0) = 0 \) where \( 0 \) is the zero vector of appropriate dimension. Let \( w \) is a global optima of \( f \). Then, we have

\[
f(\hat{w}) \leq f(w_0) \leq f(\hat{w}) + \frac{2}{\epsilon} \langle -\nabla_w f(w_0), \hat{w} - w_0 \rangle
\]

Note that the first inequality follows from the fact that \( \hat{w} \) is a global optima and second inequality follows from Lemma 1. This completes the proof.

**Remark 1** To hold the statement of Theorem 1, the gradient at a point is required to exist. However, it can be easily shown that if the gradient at a point \( w_0 \) does not exist, but the vector \( 0 \) belongs to the complement set of the interior of the tangent cone of the risk \( \hat{R}_{\epsilon,c}(w) \) at \( w_0 \), then also \( w_0 \) is a global optimum.

Recently in (Laurent & Brecht, 2018), Laurent et al. have shown that all the local optima of a deep linear network with arbitrary convex differentiable loss function are also global optima. Their results along with Theorem 1 directly imply the following corollary.

**Corollary 1** If gradient exists at all the local optima and all the hidden layers are at least as wide as either the input layer or the output layer, then all the local optima of the deep linear network with \( \epsilon \)-exponentiated loss function are also global optima.

In the next section, we derive a novel upper bound for generalization error. Using the derived bound, we show that the empirical risk minimization with \( \epsilon \)-exponentiated loss can have tighter bound than with the corresponding convex loss in spite of the fact that the Lipschitz constant for the \( \epsilon \)-exponentiated loss can be considerably larger (in the order of \( c^{\epsilon-1} \times L_{\epsilon} \) where \( L_{\epsilon} \) is the Lipschitz constant of the corresponding convex loss) than that of corresponding convex loss when \( c \ll 1 \).

### 3. Generalization Error Bounds of Empirical Risk Minimization with \( \epsilon \)-Exponentiated Loss

In this section, we present an upper bound for the generalization error incurred by an \( \epsilon \)-exponentiated loss function. Towards this end, we first propose a novel method for estimating the upper bound. Our introduced method of generalization error bound captures the average behaviour of a loss function as opposed to other existing methods (Rosasco et al., 2004) which captures the worst case behaviour. More particularly, our method is more suitable for analysing non-convex problems where the risk function is smooth in most of the regions but contains some very low probable high gradient regions. Consequently, our bound shows an weak dependence on the Lipschitz constant of the loss functions as opposed to other existing methods (Rosasco et al., 2004) which depend on the Lipschitz constant monotonically. Finally, applying the derived bound, we show that empirical risk minimization with \( \epsilon \)-exponentiated loss function can have tighter generalization error bound than that can be obtained using the corresponding convex loss function.

#### 3.1. Upper Bound for the Generalization Error

The gradient of an \( \epsilon \)-exponentiated loss function can be very large (in the order of \( c^{\epsilon-1} \times L_{\epsilon} \) where \( L_{\epsilon} \) is the Lipschitz constant of the corresponding convex loss) making the Lipschitz constant of the transformed loss very large for \( c \ll 1 \). On the other hand, the existing generalization error bound gets loose as the Lipschitz constant gets larger. To overcome this issue, we propose a novel bound for the same. Our bound is based on the work of (Rosasco et al., 2004). Before stating our bound, let us introduce certain notations and definitions.

**Definition 1** A function \( f : A \mapsto \mathbb{R}, A \subseteq \mathbb{R}^n \) is said to be
\[ L_f \text{-Lipschitz continuous, } L_f > 0, \text{ if } \] 
\[ |f(a) - f(b)| \leq L_f ||a - b||_2 \] (5)

for every \( a, b \in \mathcal{A} \).

**Definition 2** A function \( f : A \to \mathbb{R}, \mathcal{A} \subseteq \mathbb{R}^n, \) is said to be Lipschitz in the small continuous, if there exists \( \epsilon > 0 \) and \( L_f(\epsilon) > 0 \) such that
\[ ||a - b||_2 \leq \epsilon \text{ implies } |f(a) - f(b)| \leq L_f(\epsilon)||a - b||_2 \] (6)

for every \( a, b \in \mathcal{A} \).

Note that, in general, whenever a function \( f(x) \) is continuous and differentiable, \( L_f(\epsilon) \geq \sup_x |f'(x)| = L_f \) for all \( \epsilon > 0 \) where \( f'(x) \) is the gradient of \( f(x) \) at \( x \). However, this might not be true when the function \( f(x) \) also depends on the distribution of the input \( x \).

With the above definitions, we state our generalization error bound in the next theorem. Note that since a close ball in \( \mathbb{R}^d \) defined as \( \mathbb{W}_M \triangleq \{w \in \mathbb{R}^d ||w||_2 \leq M\} \) is a compact set, we can cover the set by taking union of a finite number of balls of radius \( \epsilon \) for any \( \epsilon > 0 \). Let us denote the covering number of \( \mathbb{W}_M \) by \( C(\epsilon) \). Also, we define the expected risk corresponding to the empirical risk given by Eq. (2)
\[ \mathbb{R}_L(w) = \mathbb{E}_{x,y}[l(w^T \phi(x), y)] \] (7)

where \( \mathbb{E}_{x,y}[\cdot] \) denotes expectation over the joint distribution of \( x \) and \( y \). Also note that so far we have used the notation \( l(\cdot, \cdot) \) to represent a convex loss function. However, in this section, we use the notation to represent any arbitrary loss function. With the above definitions and notations, we state our generalization error bound in the following theorem.

**Theorem 2** Let \( \mathbb{D}_N = (x_i, y_i)_{i=1}^N \) such that \( \phi(x_i) \in \{\phi(x) \in \mathbb{R}^d ||\phi(x)||_2 \leq 1\}, \) and \( y_i \in \{-1, +1\} \). Let \( \mathbb{W}_M \triangleq \{w \in \mathbb{R}^d ||w||_2 \leq M\} \) with \( M \geq 1 \). Let the loss function \( l(\cdot, \cdot) \) is \( L_1 \)-Lipschitz continuous. Set \( B = L_{R_1}(M)M + C_{l} \) where \( L_{R_1}(\epsilon) \) is as defined in Eq. (6) and \( C_l > 0 \) such that \( C_l \geq l(0, y) \) for \( y \in \{-1, +1\} \). Then for all \( \epsilon > 0 \), we have
\[ P \left( \left\{ \mathbb{D}_N \bigg| \sup_{w \in \mathbb{W}_M} |\mathbb{R}_L(w) - \mathbb{R}_L(w; \mathbb{D}_N)| \leq \epsilon + \frac{L_1 \epsilon^2}{2B} \right\} \right) \]
\[ \geq 1 - 2C \left( \frac{\epsilon}{4L_{R_1}(\epsilon')} + 1 \right) \exp \left( -\frac{N\epsilon^2}{8B^2} \right), \] (8)

where \( \epsilon' > 0 \) such that \( \epsilon' \geq \min \left\{ \epsilon, \frac{\epsilon}{2L_{R_1}(\epsilon')} \right\} \), (and this always exists).

Proof of Theorem 2 has been skipped to Appendix B. To compare our result with the previous result, we state the result of (Rosasco et al., 2004) in the next theorem:

**Theorem 3** (Rosasco et al., 2004) Let \( \mathbb{D}_N, M, \mathbb{W}_M, L_1 \) and \( C_l \) are as defined in Theorem 2. Set \( B = L_{1}M + C_l \). Then for all \( \epsilon > 0 \), we have
\[ P \left( \left\{ \mathbb{D}_N \bigg| \sup_{w \in \mathbb{W}_M} |\mathbb{R}_L(w) - \mathbb{R}_L(w; \mathbb{D}_N)| \leq \epsilon \right\} \right) \]
\[ \geq 1 - 2C \left( \frac{\epsilon}{4L_{1}C_{l}} \right) \exp \left( -\frac{N\epsilon^2}{8B^2} \right). \] (9)

**Remark 2** The confidence bound in the RHS of Eq. (9) involves \( L_1 \) the Lipschitz constant of the loss function. Thus, the bound is a monotonically decreasing function of \( L_1 \) i.e., it gets worse as \( L_1 \) gets larger. On the other hand, the confidence bound of Eq. (8) no more involve the Lipschitz constant of the loss function \( L_1 \). Instead, it involves \( L_{R_1}(\epsilon) \) which can be reasonably small even when \( L_1 \) is very large.

**Remark 3** By comparing Eq.(8) and (9), we see that there are two main differences. First, in LHS of Eq. (8), \( \epsilon \) has been replaced by a slightly larger quantity \( \epsilon + L_1 \epsilon^2/2B \). Since we generally take \( \epsilon \ll 1 \) and \( B \geq 1 \), \( L_1 \epsilon^2/2B \) can be a negligible quantity even for reasonably large \( L_1 \). Thus, it does not compromise the error bound significantly. Secondly, in RHS Eq. (8), \( C(\epsilon/4L_{1}C_{l}) \) has been replaced by \( C(\epsilon/4L_{R_1}(\epsilon')) \) + 1. Since for \( x \ll 1 \), the covering number \( C(x) \gg 1 \), Eq. (8) also does not compromise the confidence probability significantly. Moreover, if \( L_{R_1}(\epsilon') \) is reasonably smaller than \( L_1 \), the confidence bound given by Eq. (8) can be significantly better than that given by Eq. (9).

**Remark 4** For the nonconvex problem where the risk is smooth on most of the regions in its domain but has very high gradient on some very low probable regions, the bound given by Theorem 3 can be very loose as the corresponding Lipschitz constant can be very large. However, Theorem 2 can still provides a tight bound under proper distributional assumption. Thus, Theorem 2 is better suitable for analyzing nonconvex problems.

### 3.2. Comparison of Generalization Error Bound

From Theorem 2, we see that when \( L_1 \epsilon/2B \ll 1 \), the generalization error bound is a monotonically decreasing function of \( L_{R_1}(\epsilon) \) where \( l(\cdot, \cdot) \) is the loss function used in the empirical risk minimization. Thus, to compare the generalization error bound of an \( \epsilon \)-exponentiated loss function with that of the corresponding convex loss function, we compare \( L_{R_1}(\epsilon) \) with \( L_{R_{1\text{-exp}}}(\epsilon) \) where \( l(\cdot, \cdot) \) is a convex loss function and \( l^{\epsilon-\text{exp}}(\cdot, \cdot) \) is its \( \epsilon \)-exponentiated transformation. Since \( L_{R_1}(\epsilon) \) depends on the distribution \( x \) and \( y \), we assume that the margin \( y\hat{y} = yw^T \phi(x) \) follows an uniform distribution. Moreover, since by our previous assumptions, \( ||\phi(x)||_2 \leq 1 \) and \( ||w||_2 \leq M \), \( y\hat{y} \leq M \). Note that in this case,
\[ L_{R_{1\text{-exp}}}(\epsilon) = L_{R_{1\text{-exp}}}(M) = \sup_{||w||_2 \leq M} \left| \frac{d}{dw} R_{l^{\epsilon-\text{exp}}} \right|_2 = L_{R_{1\text{-exp}}}. \]
Thus, we compute an upper bound of $L_{R_1,e,c}$ as

$$L_{R_1,e,c} = \sup_{\|w\|_2 \leq M} \left\| \frac{d}{dw} \mathbf{E}_{x,y} \left[ l(\sigma^{e,c}(w^T \phi(x)), y) \right] \right\|_2$$

$$= \sup_{\|w\|_2 \leq M} \left\| E_{x,y} \left[ \frac{d}{dw} l(\sigma^{e,c}(w^T \phi(x)), y) \right] \right\|_2$$

$$\equiv \left| E_{-M \leq \delta \leq M} \left[ \frac{d}{d\delta} l(\sigma^{e,c}(\delta)) \right] \right| \text{ where } \delta = \hat{y} \tag{10}$$

The RHS of Eq. (10) can be shown to be less than $L_{R_1} \equiv |E_{-M \leq \delta \leq M} \left[ \frac{d}{d\delta} l(\delta) \right]|$ for sufficiently large $M$ and convex loss function $l(\cdot, \cdot)$ with non-positive gradient. Note that most of the standard convex loss functions for classification have gradient which is non-positive.

In the next section, we show the experimental results using $e$-exponentiated loss functions.

### 4. Experimental Results

To demonstrate the improvement obtained using $e$-exponentiated loss functions empirically, we show the results of two sets of experiments. In the first set of experiments, we have compared the accuracies obtained using $e$-exponentiated loss function with that obtained using the corresponding convex loss function on a subset of ImageNet dataset (Deng et al., 2009). In the second set of experiments, we compared the $e$-exponentiated loss functions with other state of the art methods for noisy label learning on four datasets.

#### 4.1. Experiments on ImageNet Dataset

To show the improvement in accuracies using the $e$-exponentiated loss functions over the corresponding convex loss functions, we have performed experiments on a subset of ImageNet dataset. Our collected subset of ImageNet dataset contains 511, 544 images of 1000 labels. We have randomly splitted the dataset into training set of 400,000, validation set of 50,000 and test set of 61,544 images. For the experiments, we have extracted pre-trained features of the images by passing them through the first five layers of a pre-trained AlexNet model (Krizhevsky et al., 2012). We have downloaded the pre-trained model from (Shelhamer, 2013 (accessed October, 2018) and use the code of (Kratzert, 2017 (accessed October, 2018) for extracting the pre-trained features. Note that there are only 223 labels common in between our subset of ImageNet dataset and ImageNet LSVR-C-2010 contest dataset on which the AlexNet model has been pre-trained.

For classification using the pre-trained features, we have used a three layer fully connected neural network with ReLU activation. We performed the experiments using the $e$-exponentiated softmax loss and logistic loss by varying $e = 1, 0.75$ and $0.60$ and setting $c = 0$. Note that $e = 1$ gives us back the original convex loss function. We set the dimension of the hidden layers to be 800 and used Adam optimizer for optimization. To find the suitable value of initial learning rate and keep probability for the dropout, we performed cross-validation using the top-5 accuracy on the validation set. The top-1 and top-5 test accuracies of all the experiments are shown in Table 1. The results shows that we have got a 3 to 4% improvement in top-1 and top-5 accuracies for $e = 0.6$ over $e = 1.0$ for both softmax and logistic loss. For $e = 0.75$, the accuracies obtained are in between the accuracies obtained by $e = 0.60$ and $e = 1.0$.

| Loss function | $e$ | Top-1 | Top-5 |
|---------------|----|-------|-------|
| Logistic      | 0.60 | 40.84 | 67.01 |
|               | 0.75 | 39.12 | 65.66 |
|               | 1.00 | 36.95 | 63.44 |
| Softmax       | 0.60 | 39.30 | 67.01 |
|               | 0.75 | 36.00 | 63.52 |
|               | 1.00 | 35.31 | 62.88 |

Table 1. Top-1 and Top-5 accuracies obtained on subset of ImageNet dataset. We have used $e$-exponentiated logistic and softmax loss function. Experiments are performed using $e = 1, 0.75$ and $0.60$. Note that $e$-exponentiated loss function with $e = 1$ gives us back the original convex loss function.

#### 4.2. Comparison with Other State-of-the-art Methods for Noisy Label Learning

In this section, we compare the accuracies obtained using $e$-exponentiated loss function with other state-of-the-art methods by adding label noise on the training set. For the purpose, we have adopted the experimental setup of (Ma et al., 2018).

**Experimental Setup** As in (Ma et al., 2018), we performed the experiments by adding $0\%$, $20\%$, $40\%$ and $60\%$ symmetric label noise on four benchmark datasets: MNIST ((LeCun et al., 1998)), SVHN ((Netzer et al., 2011)), CIFAR-10 ((Krizhevsky, 2009)) and CIFAR-100 ((Krizhevsky, 2009)). For all the datasets, we have used the same model and optimization setup as used in (Ma et al., 2018). Additionally, we have performed experiments using $e$-exponentiated softmax loss function with $c = 0.005$ and varying $e = 1.0, 0.75$ and 1.0. As mentioned earlier, $e = 1$ gives us back the corresponding softmax loss. Following Ma et al. in (Ma et al., 2018), we have repeated the experiments five times and reported the mean accuracies.

**Baseline Methods** For the comparison purpose, we have used the baseline methods which have been used in (Ma et al., 2018).
et al., 2018). For the shake of completeness, we briefly describe those:

**Forward (Patrini et al., 2017)** Noisy labels are corrected by multiplying the network predictions with a label transition matrix.

**Backward (Patrini et al., 2017)** Noise labels are corrected by multiplying the loss by the inverse of a label transition matrix.

**Boot-soft (Reed et al., 2014)** Loss function is modified by replacing the target label by a convex combination of the target label and the network output.

**Boot-hard (Reed et al., 2014)** It is same as Boot-soft except that instead of directly using the class predictions in the convex combination, it converts the class prediction vector to a \{0, 1\}-vector by thresholding before using in the convex combination.

**D2L (Ma et al., 2018)** It uses an adaptive loss function which exploits the differential behaviour of the deep representation subspace while a network is trained on noisy labels.

**Training with \(e\)-Exponentiated Loss function** We have found that for larger network, the rate of convergence using \(e\)-exponentiated loss function in the initial iterations are slow due to smaller magnitude of gradients. For a similar problem, Barron et al., in (Barron, 2017), have used an “annealing” approach in which, at the beginning of the optimization, they start with a convex loss function and at each epoch they gradually make the loss function non-convex by slowly tuning a hyper-parameter. However, in our experiments, we take a simpler approach. For the first total\_epoch/10 epochs, where total\_epoch is the total number of epochs the model is trained, we trained the model by setting \(e = 1\). After total\_epoch/10 epochs, we switch the value of \(e\) to our desired lower value. We take it as a future work to use a more sophisticated approaches like “annealing” in our experiments.

**Results** The results are shown in Table 2. From the table, we can see that the accuracies obtained by \(e\)-exponentiated softmax loss with \(e = 0.6\) are comparable (within the 1\% margin) or better 12 out of 16 times for methods Backward, Boot-hard and 15 out of 16 times for method Boot-soft. However, its performance is relatively worse than that of the methods Forward and D2L, in which cases the accuracies obtained by \(e\)-exponentiated loss function are comparable or better only 7 out of 16 times. Moreover, in some setting, the accuracy obtained by the two methods is better than that obtained by \(e\)-exponentiated loss function by a wide margin. However, it should be noted that the scope of our work is to develop better loss functions for the problem and many of the other label correction methods can be used along with our proposed loss functions.

**5. Conclusion**

In this paper, we have proposed \(e\)-exponentiated transformation of loss function. We have shown that empirical risk minimization with \(e\)-exponentiated convex loss function possesses the following desirable properties: 1) for a single layer network, all the stationary points of the empirical risk are global minima, and 2) for a deep linear network with all the hidden layers are at least as wide as either the input layer or the output layer, all the local minima are also global minima if the empirical risk is differentiable at all the local minima. To the best of our knowledge, this work is the first to introduce nonconvex loss function with above properties. Moreover, the \(e\)-exponentiated convex loss functions are almost differentiable, thus can be optimized using gradient descend based algorithm and more robust to outliers. Additionally, using a novel generalization error bound, we have shown that the bound can be tighter for an \(e\)-exponentiated loss function than that for the corresponding convex loss function in spite of having a much larger Lipschitz constant. Finally, by empirical evaluation, we have shown that the accuracy obtained using \(e\)-exponentiated loss function can be significantly better than that obtained using the corresponding convex loss function and comparable to the accuracy obtained by some other state of the art methods in the presence of label noise.

**References**

Barron, J. T. A more general robust loss function. *CoRR*, abs/1701.03077, 2017. URL http://arxiv.org/abs/1701.03077.

Denchev, V. S., Ding, N., Vishwanathan, S. V. N., and Neven, H. Robust classification with adiabatic quantum optimization. In *ICML*, 2012.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Ding, N. and Vishwanathan, S. V. N. t-logistic regression. In *NIPS*, pp. 514–522, 2010.

Ghosh, A., Manwani, N., and Sastry, P. S. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.

Kratzert, F. Finetune AlexNet with Tensorflow 1.0, 2017 (accessed October, 2018). URL https://github.com/kratzert/finetune_alexnet_with_tensorflow/tree/5d751d62eb4d7149f4e3f4d465febf8f07d4cea9d.
Table 2. Experiments on four benchmark datasets. For $e$-exponentiated loss functions, we have evaluated $e$-exponentiated softmax loss function with three different value of $e = 1.0$, 0.75 and 0.60. The accuracies of other methods have been taken from (Ma et al., 2018).

| Dataset  | Noise Rate | Forward | Backward | Boot-hard | Boot-soft | D2L | Softmax Crossentropy |
|----------|------------|---------|----------|-----------|-----------|-----|----------------------|
| MNIST    | 0%         | 99.30   | 99.23    | 99.13     | 99.20     | 99.28| 99.28 99.30 99.30   |
|          | 20%        | 99.45   | 99.12    | 87.69     | 88.50     | 98.84| 88.29 88.76 89.16   |
|          | 40%        | 94.90   | 70.89    | 69.49     | 70.19     | 98.49| 68.70 69.18 71.93   |
|          | 60%        | 82.88   | 52.83    | 50.45     | 46.04     | 94.73| 46.12 46.39 49.23   |
| SVHN     | 0%         | 90.22   | 90.16    | 89.47     | 89.26     | 90.32| 91.09 91.02 91.07   |
|          | 20%        | 85.51   | 79.61    | 81.21     | 79.26     | 87.63| 78.99 79.03 78.28   |
|          | 40%        | 79.09   | 64.15    | 63.25     | 64.30     | 82.68| 61.43 61.15 60.26   |
|          | 60%        | 62.57   | 53.14    | 47.61     | 39.21     | 80.92| 39.17 39.23 38.73   |
| CIFAR-10 | 0%         | 90.27   | 89.03    | 89.06     | 89.46     | 89.41| 90.33 90.36 90.17   |
|          | 20%        | 84.61   | 79.41    | 81.19     | 79.21     | 85.13| 82.00 82.94 84.70   |
|          | 40%        | 82.84   | 74.69    | 76.07     | 73.81     | 83.36| 75.60 75.86 78.62   |
|          | 60%        | 72.41   | 45.42    | 70.57     | 68.12     | 72.84| 67.02 68.36 72.35   |
| CIFAR-100| 0%         | 68.54   | 68.48    | 68.31     | 67.89     | 68.60| 68.56 68.34 67.47   |
|          | 20%        | 60.25   | 58.74    | 58.49     | 57.32     | 62.20| 59.84 61.08 61.96   |
|          | 40%        | 51.27   | 45.42    | 44.41     | 41.87     | 52.01| 51.56 53.05 54.27   |
|          | 60%        | 41.22   | 34.49    | 36.65     | 32.29     | 42.27| 38.71 39.41 39.56   |

Krizhevsky, A. Learning multiple layers of features from tiny images. University of Toronto, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.

Laurent, T. and Brecht, J. Deep linear networks with arbitrary loss: All local minima are global. In ICML, pp. 2908–2913, 2018.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Long, P. M. and Servedio, R. A. Random classification noise defeats all convex potential boosters. In ICML, pp. 608–615, 2008.

Long, P. M. and Servedio, R. A. Random classification noise defeats all convex potential boosters. volume 78, pp. 287–304, 2010.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S., Wijewickrema, S. N. R., and Bailey, J. Dimensionality-driven learning with noisy labels. In ICML, pp. 3361–3370, 2018.

Manwani, N. and Sastry, P. S. Noise tolerance under risk minimization. IEEE Trans. Cybernetics, 43(3):1146–1151, 2013.

Masnadi-Shirazi, H. and Vasconcelos, N. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In NIPS, pp. 1049–1056, 2008.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning, 2011.

Patrini, G., Rozza, A., Menon, A. K., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In CVPR, pp. 2233–2241, 2017.

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. CoRR, abs/1412.6596, 2014. URL http://arxiv.org/abs/1412.6596.

Rooyen, B., Menon, A. K., and Williamson, R. C. Learning with symmetric label noise: The importance of being unhinged. In NIPS, pp. 10–18, 2015.

Rosasco, L., Vito, E. D., Caponnetto, A., Piana, M., and Verri, A. Are loss functions all the same? volume 16, pp. 1063–107, 2004.

Shelhamer, E. bvlc_alexnet.caffemodel, 2013 (accessed October, 2018). URL http://dl.caffe.berkeleyvision.org/bvlc_alexnet.caffemodel.

A. Proof of Lemma 1
To simplify the notations, let us denote $o_{x} = \sigma^{e-c}(\hat{y})$ and $o_{l} = l^{e-c}(\hat{y}, y)$. Using the above notations, we can bound
the following inner product as:

$$\left\langle -\left[ \nabla_w l(\sigma^{c,e}(w^T \phi(x)), y) \right]_{w=w_1}, w_2 - w_1 \right\rangle$$

$$= - \left[ \frac{\partial l}{\partial \sigma} \frac{\partial \sigma}{\partial y} \right]_{\hat{y}=\hat{y}_1} \phi(x), w_2 - w_1,$$

where \( \hat{y} = w^T \phi(x) \) and \( \hat{y}_1 = w_1^T \phi(x) \)

$$= - \left[ \frac{\partial l}{\partial \sigma} \right]_{\hat{y}=\hat{y}_1} (\hat{y}_2 - \hat{y}_1), \quad \text{where} \quad \hat{y}_2 = w_1^T \phi(x)$$

$$= - \left[ \frac{\partial l}{\partial \sigma} \right]_{\hat{y}=\hat{y}_1} (\hat{y}_2 - \hat{y}_1), \quad \text{where} \quad \sigma = \sigma^{c,e}(\hat{y}_1)$$

$$= - \left[ \frac{\partial l}{\partial \sigma} \right]_{\hat{y}=\hat{y}_1} (2\sigma - 1) \left( \frac{\partial \sigma}{\partial y} \right)_{\hat{y}=\hat{y}_1} (\hat{y}_2 - \hat{y}_1)$$

\[\leq (l(\sigma_1) - l(\sigma_2)) \left( \frac{\partial \sigma}{\partial y} \right)_{\hat{y}=\hat{y}_1} (\hat{y}_2 - \hat{y}_1) \quad (11)\]

The last step follows from the convexity property of function \( l(\cdot) \) with respect to its first argument. By definition of \( \sigma^{c,e}(\cdot) \), we have \( i\sigma = \sigma^{c,e}(\hat{y}_i) \) for \( i = 1, 2 \). Thus we simplify the second factor of the RHS of Eq. (11) as

\[\left[ \frac{\partial \sigma}{\partial y} \right]_{\hat{y}=\hat{y}_1} (\hat{y}_2 - \hat{y}_1) \]

\[\geq e/2 \quad (12)\]

Putting the above lower bounds into the RHS of Eq. (11), we get

$$\left\langle -\left[ \nabla_w l(\sigma^{c,e}(w^T \phi(x)), y) \right]_{w=w_1}, w_2 - w_1 \right\rangle \geq \frac{e}{2} (l'(\sigma_1) - l'(\sigma_2))$$

\[\geq \frac{e}{2} (l'(\sigma_1) - l'(\sigma_2)) \quad (13)\]

Therefore, we can get the lower bound of LHS of Eq. (4) as

$$- \nabla_w f(w_1), w_2 - w_1$$

$$= \left\langle -\frac{1}{N} \sum_{i=1}^N [\nabla_w l(\sigma^{c,e}(w^T \phi(x)), y_i)]_{w=w_1}, w_2 - w_1 \right\rangle$$

$$= \frac{1}{N} \sum_{i=1}^N \left\langle -[\nabla_w l(\sigma^{c,e}(w^T \phi(x)), y_i)]_{w=w_1}, w_2 - w_1 \right\rangle$$

\[\geq \frac{1}{2} e (f(w_1) - f(w_2)) \quad (13)\]

This completes the proof.

**B. Proof of Theorem 2**

Before going to the proof of Theorem 2, we will state and prove another result which is required for the proof.

**Lemma 2** Let the expected risk \( R_l(w) \) be Lipschitz in small continuous and the corresponding loss function is \( L_l \)-Lipschitz. Then for \( \|w_1 - w_2\| \leq \epsilon \), \( \epsilon > 0 \), and \( \rho > 0 \)

$$|R_l(w_1) - R_l(w_2)| \leq L_{R_l}(\epsilon) \|w_1 - w_2\| + \rho \quad (14)$$

is satisfied with probability at least \( 1 - 2 \exp \left( -\frac{N\epsilon^2}{2L_{R_l}(\epsilon)} \right) \).

**Proof:** Since \( R_l(w) \) is Lipschitz in small continuous and \( \|w_1 - w_2\| \leq \epsilon \), we have

$$|R_l(w_1) - R_l(w_2)| \leq L_{R_l}(\epsilon) \|w_1 - w_2\| \leq (\epsilon) \quad (15)$$

If we let \( z_i = l(w_1^T \phi(x_i), y_i) - l(w_2^T \phi(x_i), y_i) \), then we can write

$$E[z] = R_l(w_1) - R_l(w_2), \quad \text{and} \quad \sum_{i=1}^N z_i = R_l(w_1) - R_l(w_2)$$

Since \( \|w_1 - w_2\| \leq \epsilon \) and the loss function \( l(\cdot) \) is \( L_l \)-Lipschitz function, \( |z_i| \leq L_l \epsilon \). Using Hoeffding’s inequality, we get

$$P \left\{ \left| \sum_{i=1}^N [l(w_1^T \phi(x_i), y_i) - l(w_2^T \phi(x_i), y_i)] \right| \geq \rho \right\} \leq 2 \exp \left( -\frac{N\rho^2}{2L_l(\epsilon)^2} \right) \quad (16)$$

Combining Eq. (15) and (16), we complete the proof. \( \square \)

Now we prove Theorem 2.

**Proof of Theorem 2:** We will mainly follow the proof of (Rosasco et al., 2004). For simplifying the notation, we ignore the subscript of \( D_N \), \( R_l(\cdot) \) and \( R_l(\cdot) \) through out the proof. First of all, by denoting

$$\Delta_D(w) = R(w) - \hat{R}(w) \quad (17)$$

and using Lemma 2, we get

$$|\Delta_D(w_1) - \Delta_D(w_2)| \leq |R(w_1) - R(w_2)| + |\hat{R}(w_1) - \hat{R}(w_2)| \leq 2L_{R_l}(\epsilon) \|w_1 - w_2\| + \rho \quad (18)$$

holds for all \( \|w_1 - w_2\| \leq \epsilon' \) for some \( \epsilon' > 0 \) with probability at least \( 1 - 2 \exp \left( -\frac{N\epsilon^2}{2L_{R_l}(\epsilon)} \right) \). Putting \( \rho = \frac{L_l \epsilon'}{B} \) into the above statement, we get

$$|\Delta_D(w_1) - \Delta_D(w_2)| \leq 2L_l(\epsilon') \|w_1 - w_2\| + \frac{L_l \epsilon'}{B} \quad (19)$$

with probability at least \( 1 - 2 \exp \left( -\frac{N\epsilon^2}{2B^2} \right) \). Again, in (Rosasco et al., 2004), Rosasco et al. have shown that

$$P(A) = P \left( \bigcup_{i=1}^m A_{w_i} \right) \leq 2m \exp \left( -\frac{N\epsilon^2}{2B^2} \right) \quad (20)$$
where \( w_1, \ldots, w_m \) be the \( m = C \left( \frac{\epsilon}{2L_R(\epsilon')} \right) \) points such that the close balls \( B \left( w_i, \frac{\epsilon}{2L_R(\epsilon')} \right) \) with radius \( \frac{\epsilon}{2L_R(\epsilon')} \), center \( w_i \) covers the whole set \( W_M = \{ w \in \mathbb{R}^d | ||w||_2 \leq M \} \) and

\[
A_{w_i} = \{ D | |\Delta_D(w_i)| \geq \epsilon \} \text{ for } i = 1, \ldots, m.
\]

(21)

When \( \epsilon' \geq \frac{\epsilon}{2L_R(\epsilon')} \), for all \( w \in W_M \), there exists some \( i \in \{ 1, \ldots, m \} \) such that \( w \in B \left( w_i, \frac{\epsilon}{2L_R(\epsilon')} \right) \) i.e.

\[
||w - w_i||_2 \leq \frac{\epsilon}{2L_R(\epsilon')}
\]

(22)

Note that \( D \in A \) is the dataset for which there exists some \( w_i \) whose empirical risk has not converged to its expected risk. Thus, for all \( D / \in A \), we have

\[
|\Delta_D(w) - \Delta_D(w_i)| \leq \epsilon + \frac{L_i \epsilon' \epsilon}{B}
\]

(23)

holds for all \( w \in W_M \) with probability at least \( 1 - 2 \exp \left( -\frac{N\epsilon^2}{2B^2} \right) \). Therefore, if there exists an \( \epsilon' > 0 \) such that \( \epsilon' \geq \frac{\epsilon}{2L_R(\epsilon')} \), and for all \( D / \in A \),

\[
|\Delta_D(w)| \leq 2\epsilon + \frac{L_i \epsilon' \epsilon}{B}
\]

(24)

hold with probability at least

\[
1 - 2 \exp \left( -\frac{N\epsilon^2}{2B^2} \right) \left( 1 - 2m \exp \left( -\frac{N\epsilon^2}{2B^2} \right) \right) \geq 1 - 2(m + 1) \exp \left( -\frac{N\epsilon^2}{2B^2} \right) = 1 - 2 \left( C \left( \frac{\epsilon}{2L_R(\epsilon')} \right) + 1 \right) \exp \left( -\frac{N\epsilon^2}{2B^2} \right).
\]

By replacing \( \epsilon \) with \( \epsilon/2 \) and by replacing \( \epsilon' \) by \( \epsilon \) whenever \( \epsilon' > \epsilon \), the statement of the lemma follows.

But, it still remains to show that there always exists an \( \epsilon' > 0 \) such that \( \epsilon' \geq \frac{\epsilon}{2L_R(\epsilon')} \). Note that \( L_R(\epsilon') \) is a monotonically increasing function of \( \epsilon' \). If for some \( \epsilon' < \epsilon, \epsilon' \geq \epsilon/2L_R(\epsilon') \) holds, we are already done. Else, we have \( 2\epsilon' L_R(\epsilon') < \epsilon \).

Thus, we can increase \( 2\epsilon' L_R(\epsilon') \) unboundedly by increasing \( \epsilon' \), making it larger than \( \epsilon \) eventually. \( \square \)