Max-Mahalanobis Linear Discriminant Analysis Networks

Tianyu Pang [1] **Chao Du** [1] **Jun Zhu** [1]

Abstract

A deep neural network (DNN) consists of a nonlinear transformation from an input to a feature representation, followed by a common softmax linear classifier. Though many efforts have been devoted to designing a proper architecture for nonlinear transformation, little investigation has been done on the classifier part. In this paper, we show that a properly designed classifier can improve robustness to adversarial attacks and lead to better prediction results. Specifically, we define a Max-Mahalanobis distribution (MMD) and theoretically show that if the input distributes as a MMD, the linear discriminant analysis (LDA) classifier will have the best robustness to adversarial examples. We further propose a novel Max-Mahalanobis linear discriminant analysis (MM-LDA) network, which explicitly maps a complicated data distribution in the input space to a MMD in the latent feature space and then applies LDA to make predictions. Our results demonstrate that the MM-LDA networks are significantly more robust to adversarial attacks, and have better performance in class-biased classification.

1. Introduction

Deep neural networks (DNNs) have shown state-of-the-art performance in different tasks (Goodfellow et al., 2016). A typical feed-forward DNN is a combination of a nonlinear transformation from the input x to the latent feature vector z and a linear classifier acting on z to return a prediction for x. Many different architectures for neural networks have been proposed, e.g., VGG nets (Simonyan & Zisserman, 2014), Resnets (He et al., 2016a;b) and Google nets (Szegedy et al., 2016) for powerful nonlinear transformation, while leaving the linear classifier part under-explored, which is by default defined as a softmax regression (SR) (or logistic regression (LR) for binary classification). Some work has tried to instead use linear (or kernel) SVMs as the classifier (Huang & LeCun, 2006; Ngiam et al., 2010; Coates et al., 2011; Tang, 2013). But, such techniques either do not fine-tune the lower level features w.r.t. the SVM's objective or only result in marginal improvements. Thus, SR is still the default choice, given its simplicity and smoothness.

However, the SR (or LR) classifier is not problemless. Efron (1975) shows that if the input x arises from a 2-center mixture of Gaussian distribution, then LR is less efficient than linear discriminant analysis (LDA), i.e., LR needs more training samples than LDA does to obtain a certain error rate. The relative efficiency of LR to LDA depends on the Mahalanobis distance Δ between the two Gaussian components and the log-ratio of class priors ζ. Generally, a larger value of Δ or $|\zeta|$ will lead to a lower relative efficiency of LR to LDA. Furthermore, it has been widely recognized that the DNNs with a SR classifier are vulnerable to adversarial attacks (Szegedy et al., 2014; Goodfellow et al., 2015; Nguyen et al., 2015; Moosavi-Dezfooli et al., 2016), where human imperceivable images can be crafted to fool a high-accuracy network. Though many efforts have been devoted to improving the robustness, such as using adversarial training (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017b), it still remains open on how to design a robust classifier by itself.

In this paper, we draw inspirations from Efron's analysis and design a robust classifier that is generally applicable to feedforward networks. Specifically, we define the Max-Mahalanobis distribution (MMD) for multi-class classification, which is a special mixture of Gaussian distribution. We theoretically show that if the input samples distribute as a MMD, the LDA classifier will have the best robustness to adversarial attacks. Though distributing as a MMD is not likely to hold for complex data (e.g., images in the pixel space), it may be true if we properly transform the data. Based on this result, we propose a novel Max-Mahalanobis linear discriminant analysis (MM-LDA) network, which explicitly learns a powerful nonlinear transformation network to turn the complex inputs to match the MMD in a latent feature space, and then uses the LDA principle to make predictions on the latent features. Besides robustness, since a large value of $|\zeta|$ for the data distribution indicates

[1] Dept. of Comp. Sci. & Tech., BNRist Center, State Key Lab for Intell. Tech. & Sys., THBI Lab, Tsinghua University, Beijing, 100084, China. Correspondence to: Jun Zhu <dcszj@mail.tsinghua.edu.cn>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

a high efficiency of LDA, the MM-LDA network can also perform better on *class-biased datasets*,[1] i.e., datasets with different numbers of data points for different classes, which are common in practice.

Unlike the SR classifier, whose parameters are jointly learned with those of the transformation network, the optimal parameters of the MMD are estimated separately by a simple procedure, and we only need to learn the parameters of the transformation network. The overall training objective is a cross-entropy loss. Standard training algorithms (e.g., stochastic gradient descent) are applicable with little extra computational cost. Moreover, as the MM-LDA network differs only in the classifier part, our technique can be naturally combined with different kinds of nonlinear transformation architectures (e.g., VGG nets, Resnets or Google nets) and different kinds of training methods (Liu et al., 2016; Pang et al., 2017) for good performance.

We test the proposed network on the widely used MNIST and CIFAR-10 datasets for both robustness to adversarial attacks and classification accuracy. As for robustness, we consider various adversarial attacking methods, and the results demonstrate that the MM-LDA network is indeed much more robust to adversarial examples than the SR networks, even when the SR networks are enhanced by the adversarial training methods. As for classification, we test the performance of the MM-LDA network on both class-biased and class-unbiased datasets. The results show that the MM-LDA networks can obtain higher accuracy on class-biased datasets while maintaining state-of-the-art accuracy on class-unbiased datasets.

2. Preliminary Knowledge

In this section, we first briefly introduce some notations. Then we provide a formal description of the adversarial setting and introduce some common attacking methods. Finally, we introduce the relative efficiency of logistic regression (LR) to linear discriminant analysis (LDA) in the binary-class cases, which inspires our novel network.

2.1. Notations

We refer to the DNN with a softmax output layer as an SR network, which is widely used in classification tasks (Goodfellow et al., 2016). Let L denote the number of classes ($L \geq 2$), and define $[L] = \{1, \cdots, L\}$. An SR network can be generally expressed as $F(x, \theta) = \mathbb{S}(W_s z + b_s)$, where z is the latent feature representation of the input x and the softmax function $\mathbb{S}(z) : \mathbb{R}^L \to \mathbb{R}^L$ is defined as $\mathbb{S}(z)_i = \exp(z_i)/\sum_{i=1}^{L} \exp(z_i)$ for each element $i \in [L]$. Here, θ denotes the parameters of the nonlinear transformation network from x to z. W_s and b_s are the weight matrix and bias vector of the softmax layer respectively.

2.2. The Adversarial Setting

In the adversarial setting, adversaries apply attacking methods to craft adversarial examples based on the given normal examples. We consider the *white-box attack*, which is the most challenging and difficult threat model for classifiers to defend (Carlini & Wagner, 2017b). White-box adversaries know everything, e.g., parameters, about the classifiers that they attack on. An adversarial example x^* should be indistinguishable from its normal counterpart x by human observers, but makes the classifier misclassify on it. Formally, the adversarial example x^* crafted on x should satisfy

$$\hat{y}(x^*) \neq \hat{y}(x), \text{ s. t. } \|x^* - x\| \leq \epsilon, \tag{1}$$

where $\hat{y}(\cdot)$ denotes the predicted label from the classifier, and ϵ is the maximal perturbation under a norm that varies in different attacking methods. If there is an additional constraint that $\hat{y}(x^*)$ is a specific class l, x^* is regarded as targeted. Otherwise x^* is untargeted. Let $\mathcal{L}(x, y)$ denote the training loss on (x, y). Some of the most common attacking methods are introduced below:

Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) is an one-step attacking method that the adversarial example x^* is crafted as $x^* = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y))$.

Basic Iterative Method (BIM) (Kurakin et al., 2017a) is an iterative version of FGSM. Let $x_0^* = x$, r be the number of iteration steps, then BIM crafts an adversarial example as $x_i^* = \text{clip}_{x,\epsilon}(x_{i-1}^* + \frac{\epsilon}{r} \cdot \text{sign}(\nabla_{x_{i-1}^*} \mathcal{L}(x_{i-1}^*, y)))$, where $\text{clip}_{x,\epsilon}(\cdot)$ is the clipping function.

Iterative Least-likely Class Method (ILCM) (Kurakin et al., 2017a) is a targeted version of BIM with the formula as $x_i^* = \text{clip}_{x,\epsilon}(x_{i-1}^* - \frac{\epsilon}{r} \cdot \text{sign}(\nabla_{x_{i-1}^*} \mathcal{L}(x_{i-1}^*, y_{ll})))$, where $y_{ll} = \arg\min_i F(x)_i$ is the label with minimal confidence.

Jacobian-based Saliency Map Attack (JSMA) (Papernot et al., 2016) is also a targeted attack that perturbs the feature x_i by a perturbation ϵ in each iteration step that maximizes the saliency map

$$S(x, y)[i] = \begin{cases} 0, \text{ if } \frac{\partial F(x)_y}{\partial x_i} < 0 \text{ or } \sum_{j \neq y} \frac{\partial F(x)_j}{\partial x_i} > 0, \\ (\frac{\partial F(x)_y}{\partial x_i}) \left| \sum_{j \neq y} \frac{\partial F(x)_j}{\partial x_i} \right|, \text{ otherwise.} \end{cases}$$

JSMA perturbs fewer pixels compared to other attacks.

Carlini & Wagner (C&W) (Carlini & Wagner, 2017a) defines $x^*(\omega) = \frac{1}{2}(\tanh(\omega) + 1)$ in terms of ω, and solves $\min_\omega \|x^*(\omega) - x\|_2^2 + c \cdot f(x^*(\omega))$, where c is a constant chosen by a modified binary search. Let $\mathbb{S}_{pre}(x)$ be the input vector of the softmax function in a classifier, then $f(\cdot)$ is the objective function defined as

$$f(x) = \max(\max\{\mathbb{S}_{pre}(x)_i : i \neq y\} - \mathbb{S}_{pre}(x)_i, -\kappa),$$

[1] Our setting differs from the previous work (Huang et al., 2017; Fallah et al., 2017), where only the training set is class-biased.

where κ controls the confidence on adversarial examples.

The attacking methods are generally gradient-based. They can be categorized into two groups. The first one consists of iterative-based methods, e.g., FGSM, BIM, ILCM and JSMA. These methods usually iterate less than hundreds of rounds to craft an adversarial example. Besides, they often blend the constraints into their updating operations (e.g., adding a sign function on the gradients under the L_∞ norm constraint). The second group consists of optimization-based methods, e.g., the C&W method. These methods require much more computation compared to the iterative-based methods, since optimization-based methods iterate thousands of rounds to craft an adversarial example. However, usually an optimization-based method has higher success rates on attacking classifiers.

2.3. The Relative Efficiency of LR to LDA

Softmax regression (SR) is the most commonly used model as the linear classifier part in neural networks (Goodfellow et al., 2016). In the binary-class cases, SR reduces to LR. We denote the two classes with labels 0 and 1, and make the following assumption of the input x with its label y.

Assumption 1. *The distribution of the p-dimensional random vector x with its class label y is*

$$P(y=i)=\pi_i,\, P(x|y=i)=\mathcal{N}(\mu_i,\Sigma),$$

where $i\in\{0,1\}$, $\pi_0+\pi_1=1$ and each conditional Gaussian distribution has the same covariance matrix Σ.

Efron (1975) shows that under Assumption 1, LR is asymptotically less efficient than LDA. Specifically, we denote the decision regions of a classifier as R_0 and R_1, the error rate of a classifier is defined as

$$\begin{aligned}\mathrm{ER}=\pi_0 P(x\in R_1|x\sim\mathcal{N}(\mu_0,\Sigma))\\ +\pi_1 P(x\in R_0|x\sim\mathcal{N}(\mu_1,\Sigma)).\end{aligned}$$

Then the relative efficiency of LR to LDA is defined as

$$\mathrm{Eff}_p(\zeta,\Delta)=\lim_{N\to\infty}\frac{\mathbb{E}[\mathrm{ER}_{\mathrm{LDA}}-\mathrm{ER}_{\mathrm{Bayes}}]}{\mathbb{E}[\mathrm{ER}_{\mathrm{LR}}-\mathrm{ER}_{\mathrm{Bayes}}]},$$

where $\mathrm{ER}_{\mathrm{Bayes}}$ is the Bayes error rate, N is the number of training data points and $\Delta=[(\mu_1-\mu_0)^\top\Sigma^{-1}(\mu_1-\mu_0)]^{\frac{1}{2}}$ is the Mahalanobis distance between the two conditional Gaussian components. A lower value of $\mathrm{Eff}_p(\zeta,\Delta)$ indicates that asymptotically LDA needs less training data points than LR does to obtain a certain error rate.

In order to calculate $\mathrm{Eff}_p(\zeta,\Delta)$, we let A_i be

$$A_i(\pi_0,\Delta)=\int_{-\infty}^{\infty}\frac{e^{-\Delta^2/8}x^i\varphi(x)}{\pi_0 e^{-\Delta x/2}+\pi_1 e^{\Delta x/2}}dx,$$

where $\varphi(x)=(2\pi)^{\frac{1}{2}}\exp(-x^2/2)$ is the probability density function of $\mathcal{N}(0,1)$. Then there is:

Theorem 1. *(Efron, 1975) The relative efficiency of logistic regression to linear discriminant analysis is*

$$\mathrm{Eff}_p(\zeta,\Delta)=(Q_1+(p-1)Q_2)/(Q_3+(p-1)Q_4),$$

where $Q_2=1+\pi_0\pi_1\Delta^2$, $Q_4=\frac{1}{A_0}$ and

$$Q_1=\begin{pmatrix}1 & \frac{\zeta}{\Delta}\end{pmatrix}\begin{bmatrix}1+\frac{\Delta^2}{4} & (\pi_0-\pi_1)\frac{\Delta}{2}\\ (\pi_0-\pi_1)\frac{\Delta}{2} & 1+2\pi_0\pi_1\Delta^2\end{bmatrix}\begin{bmatrix}1\\ \frac{\zeta}{\Delta}\end{bmatrix},$$

$$Q_3=\begin{pmatrix}1 & \frac{\zeta}{\Delta}\end{pmatrix}\frac{1}{A_0A_2-A_1^2}\begin{bmatrix}A_2 & A_1\\ A_1 & A_0\end{bmatrix}\begin{bmatrix}1\\ \frac{\zeta}{\Delta}\end{bmatrix}.$$

Generally, larger values of $|\zeta|$ or Δ imply lower values of $\mathrm{Eff}_p(\zeta,\Delta)$, and thus lower relative efficiency of LR to LDA.

3. Methodology

We now present our method in this section. We first define the Max-Mahalanobis distribution (MMD) with theoretical analyses, and then propose the Max-Mahalanobis linear discriminant analysis (MM-LDA) network.

3.1. Max-Mahalanobis Distribution

We consider the multi-class cases, and a natural extension of Assumption 1 is as follows.

Assumption 2. *The distribution of the p-dimensional random vector x with its class label y is*

$$P(y=i)=\pi_i,\, P(x|y=i)=\mathcal{N}(\mu_i,\Sigma),$$

where $i\in[L]$, $\sum_{i=1}^L\pi_i=1$ and each conditional Gaussian distribution has the same covariance matrix Σ.

Then, the Mahalanobis distance between any two Gaussian components i and j is $\Delta_{i,j}=[(\mu_i-\mu_j)^\top\Sigma^{-1}(\mu_i-\mu_j)]^{\frac{1}{2}}$. As suggested in Efron (1975), there is no loss of generality to assume that Σ is nonsingular. Thus we can do the Cholesky decomposition as $\Sigma=QQ^\top$, where Q is a lower triangular matrix with positive diagonal entries. By applying the linear transformation $\widetilde{x}=Q^{-1}(x-\overline{\mu})$, where $\overline{\mu}=\sum_{i=1}^L\mu_i/L$, we can reduce Assumption 2 to the standard form.

Assumption 3. *The distribution of the p-dimensional random vector x with its class label y is*

$$P(y=i)=\pi_i,\, P(\widetilde{x}|y=i)=\mathcal{N}(\widetilde{\mu}_i,\mathrm{I}),$$

where $i\in[L]$, $\sum_{i=1}^L\pi_i=1$ and $\sum_{i=1}^L\widetilde{\mu}_i=0$.

For the standard form, we have $\widetilde{\Delta}_{i,j}=[(\widetilde{\mu}_i-\widetilde{\mu}_j)^\top(\widetilde{\mu}_i-\widetilde{\mu}_j)]^{\frac{1}{2}}$. Note that the linear transformation $x\mapsto\widetilde{x}$ keeps the Mahalanobis distances invariant, i.e., $\forall i,j\in[L]$, there is $\widetilde{\Delta}_{i,j}=\Delta_{i,j}$. In the sequel, we will assume that the input pair (x,y) satisfies Assumption 3. For notation clarity, we denote $\widetilde{x}$ as x, $\widetilde{\mu}_i$ as μ_i, $\widetilde{\Delta}_{i,j}$ as $\Delta_{i,j}$ without ambiguity.

This distribution is of interest as we can explicitly characterize the robustness to adversarial samples of a LDA classifier. Specifically, the decision boundary obtained by LDA between class i and j is decided by the Fisher's linear discriminant function (Friedman et al., 2001), $\lambda_{i,j}(x) = \beta_{i,j} + \alpha_{i,j}^{\top} x = 0$, where

$$\beta_{i,j} = \log(\pi_i/\pi_j) + \frac{1}{2}(\|\mu_j\|_2^2 - \|\mu_i\|_2^2),$$
$$\alpha_{i,j}^{\top} = (\mu_i - \mu_j)^{\top}.$$

In the adversarial setting, the nearest adversarial example x^* that satisfies condition (1) w.r.t the normal example x must be located on the decision boundary (Moosavi-Dezfooli et al., 2016). We randomly sample a normal example of class i as $x_{(i)}$, i.e., $x_{(i)} \sim \mathcal{N}(\mu_i, I)$, and denote its nearest adversarial counterpart on the decision boundary $\lambda_{i,j}(x) = 0$ as $x^*_{(i,j)}$. According to condition (1), there is $\hat{y}(x_{(i)}) = i, \hat{y}(x^*_{(i,j)}) = j$ or $\hat{y}(x_{(i)}) = j, \hat{y}(x^*_{(i,j)}) = i$, where $\hat{y}(\cdot)$ refers to the predicted label from the LDA classifier. We define the distance between $x_{(i)}$ and $x^*_{(i,j)}$ as $d_{(i,j)}$. Then we have the theorem on the relationship between the expectation $\mathbb{E}[d_{(i,j)}]$ and the Mahalanobis distance $\Delta_{i,j}$:

Theorem 2. *(Proof in Appendix A) If $\pi_i = \pi_j$, the expectation of the distance $d_{(i,j)}$ is a function of the Mahalanobis distance $\Delta_{i,j}$:*

$$\mathbb{E}[d_{(i,j)}] = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\Delta_{i,j}^2}{8}\right) + \frac{1}{2}\Delta_{i,j}\left[1 - 2\Phi(-\frac{\Delta_{i,j}}{2})\right],$$

where $\Phi(\cdot)$ is the normal cumulative distribution function.

The more general result when $\pi_i \neq \pi_j$ can be found in the proof of Theorem 2, which leads to similar conclusions. Furthermore, we can show that $\mathbb{E}[d_{(i,j)}]$ monotonically increases w.r.t $\Delta_{i,j}$, as summarized in the following corollary.

Corollary 1. *The partial derivative of $\mathbb{E}[d_{(i,j)}]$ w.r.t $\Delta_{i,j}$ is*

$$\frac{\partial \mathbb{E}[d_{(i,j)}]}{\partial \Delta_{i,j}} = \frac{1}{2}[1 - 2\Phi(-\frac{\Delta_{i,j}}{2})] \geq 0,$$

where the Mahalanobis distance $\Delta_{i,j}$ is non-negative.

Moosavi-Dezfooli et al. (2016) define the robustness of a point $x_{(i)}$ as $\min_{j \neq i} d_{(i,j)}$. Similar to this definition, we define the robustness of the classifier as below. Note that $\mathbb{E}[d_{(i,j)}]$ is the expectation value of the minimal distance from a normal example to its potential adversarial counterpart between class i and j. Thus $\mathbb{E}[d_{(i,j)}]$ can measure the local robustness of the classifier on the attacks focusing on the two classes, where a larger value of $\mathbb{E}[d_{(i,j)}]$ indicates better local robustness, and vice verse. Then the robustness of the classifier on all the attacks can be measured by

$$\mathrm{RB} = \min_{i,j \in [L]} \mathbb{E}[d_{(i,j)}], \tag{2}$$

which can be regarded as a tight lower bound of the local robustness between any two classes. Because $\mathbb{E}[d_{(i,j)}]$ monotonically increases w.r.t $\Delta_{i,j}$, we prefer larger values of $\Delta_{i,j}$ for better local robustness. According to Corollary 1, the gap $\left|\mathbb{E}[d_{(i,j)}]/\Delta_{i,j} - 1/2\right|$ monotonically decreases to 0 w.r.t $\Delta_{i,j}$, e.g., when $\Delta_{i,j} > 10$, we can numerically figure out that $\left|\mathbb{E}[d_{(i,j)}]/\Delta_{i,j} - 1/2\right| < 10^{-7}$. Thus we can approximate $\mathbb{E}[d_{(i,j)}]$ using $\Delta_{i,j}/2$, which further results in an approximation for the robustness RB as

$$\mathrm{RB} \approx \overline{\mathrm{RB}} = \min_{i,j \in [L]} \Delta_{i,j}/2. \tag{3}$$

Algorithm 1 GenerateOptMeans

```
**Input:** The constant $C$, the dimension of vectors $p$ and the number of classes $L$. $(L \leq p+1)$
**Initialization:** Let the $L$ mean vectors be $\mu_1^* = e_1$ and $\mu_i^* = 0_p, i \neq 1$. Here $e_1$ and $0_p$ separately denote the first unit basis vector and the zero vector in $\mathbb{R}^p$.
for $i = 2$ to $L$ do
  for $j = 1$ to $i-1$ do
    $\mu_i^*(j) = -[1 + \langle \mu_i^*, \mu_j^* \rangle \cdot (L-1)]/[\mu_j^*(j) \cdot (L-1)]$
  end for
  $\mu_i^*(i) = \sqrt{1 - \|\mu_i^*\|_2^2}$
end for
for $k = 1$ to $L$ do
  $\mu_k^* = \sqrt{C} \cdot \mu_k^*$
end for
**Return:** The optimal mean vectors $\mu_i^*, i \in [L]$.
```

We now investigate when the approximated robustness $\overline{\mathrm{RB}}$ of the LDA classifier can achieve its maximal value, and derive an efficient algorithm to estimate the unknown means $\mu = \{\mu_i | i \in [L]\}$ of the input distribution. Let $\|\mu\|_2$ be $\max_i \|\mu_i\|_2$. Since μ has finite elements, there always exists a positive constant C, such that $\|\mu\|_2^2 = C$. The following theorem gives a tight upper bound for $\overline{\mathrm{RB}}$.

Theorem 3. *(Proof in Appendix A) Assume that $\sum_{i=1}^{L} \mu_i = 0$ and $\|\mu\|_2^2 = C$. Then we have*

$$\overline{\mathrm{RB}} \leq \sqrt{\frac{LC}{2(L-1)}}.$$

The equality holds if and only if

$$\mu_i^{\top} \mu_j = \begin{cases} C, & i = j, \\ C/(1-L), & i \neq j, \end{cases} \tag{4}$$

where $i, j \in [L]$ and $\mu_i, \mu_j \in \mu$.

We denote any set of means that satisfy the optimal condition (4) as μ^*. When $L \leq p+1$,[2] there is an infinite

[2] Otherwise, there is no solution for μ^*.

number of μ^* because of the degeneracy of the condition. Intuitively, the elements in μ^* constitute the vertexes of an equilateral triangle when $L = 3$, and those of a regular tetrahedron when $L = 4$. In Alg. 1, we propose an easy-to-implement method to construct a set of means μ_0^* that satisfy the condition, where $\mu_0^* = \mathrm{GenerateOptMeans}(C, p, L)$.

With the above results, we formally define a joint distribution with the form

$$P(y = i) = \pi_i,\, P(x|y = i) = \mathcal{N}(\mu_i^*, \mathrm{I}),\, i \in [L]$$

as a *Max-Mahalanobis distribution (MMD)*, namely, it has the maximal minimal Mahalanobis distance between any two Gaussian components for a given $\|\mu\|_2$. In a nutshell, when regarding the set of means μ in Assumption 3 as independent variables for a given $\|\mu\|_2$, the LDA classifier would have the best robustness if its input distributes as a MMD. We refer to the above process as the *Max-Mahalanobis linear discriminant analysis (MM-LDA)* procedure.

3.2. The MM-LDA Network

Though elegant, the MM-LDA procedure is not directly applicable in practice, as the mixture of Gaussian Assumption 3 is unlikely to hold in the input space (e.g., images in the pixel space), in which the data distribution $P(x)$ can be very complex. Fortunately, thanks to the universal approximation power of neural networks (Hornik et al., 1989) and the algorithmic advances, previous work on deep generative models (Goodfellow et al., 2014; Kingma & Welling, 2013) has demonstrated that a deep neural network can be learned to transform a simple distribution (e.g., standard normal) to a complex one that matches the data distribution. The reverse direction is also true, and it has been implicitly applied in feed-forward networks, where a powerful nonlinear transformation network is learned to turn a complexly distributed input data into a latent feature space, and then a linear classifier (e.g., SR) is sufficient to achieve state-of-the-art performance (Simonyan & Zisserman, 2014; He et al., 2016a;b; Szegedy et al., 2016). Therefore, we can expect that the MM-LDA procedure will work well on a properly learned latent feature space by exploring the power for DNNs, as detailed below.

Formally, we propose the *Max-Mahalanobis linear discriminant analysis (MM-LDA)* network, which consists of a nonlinear transformation network (characterized as a DNN) to turn the input x into a latent feature representation z, and applies the MM-LDA procedure on z. Namely, the MM-LDA network explicitly models $P(z)$ as a MMD, and applies LDA on z to make predictions. According to the analysis in Section 3.1, the MM-LDA network can have the best robustness in the latent feature space, and further results in better robustness in the input space.

Algorithm 2 The training phase for the MM-LDA network

```
Input: The model $z_\theta(x)$, the square norm $C$ of Gaussian means, the training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i \in [N]}$.
Initialization: Initialize $\theta$ as $\theta_0$, the training step as $s = 0$. Let $p = \dim(z)$, $\varepsilon$ be the learning rate variable.
Get $\mu^* = \mathrm{GenerateOptMeans}(C, p, L)$ for the MMD.
while not converged do
    Sample a mini-batch of training data $\mathcal{D}_m$ from $\mathcal{D}$,
    Calculate the objective
        $$\mathcal{L}_{\mathrm{MM}}^m = \frac{1}{|\mathcal{D}_m|} \sum_{(x_i, y_i) \in \mathcal{D}_m} \mathcal{L}_{\mathrm{MM}}(x_i, y_i, \mu^*),$$
    Update parameters $\theta_{s+1} \leftarrow \theta_s - \varepsilon \nabla_\theta \mathcal{L}_{\mathrm{MM}}^m$,
    Set $s \leftarrow s + 1$.
end while
Return: The parameters $\theta_{\mathrm{MM}} = \theta_s$.
```

Given a feature vector z in the MM-LDA network, according to Bayes' theorem and the definition of MMD, we have the conditional distribution of labels:

$$P(y{=}k|z) = \frac{P(z|y{=}k)P(y{=}k)}{P(z)} = \frac{\pi_k \mathcal{N}(z|\mu_k^*, \mathrm{I})}{\sum_{i=1}^{L} \pi_i \mathcal{N}(z|\mu_i^*, \mathrm{I})}.$$

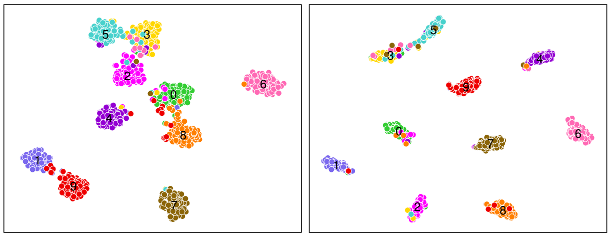
Note that the feature vector z is actually $z_\theta(x)$, since it is obtained by the nonlinear transformation network $x \mapsto z$, parameterized by θ. Instead of estimating model parameters from data as LDA does, MM-LDA treats the set of means μ^* and different class priors $\pi_i, i \in [L]$ as hyperparameters, and θ is what the MM-LDA network needs to learn in the training phase. Similar to the SR network, we let $F_{\mathrm{MM}}(x)$ be the output prediction of the MM-LDA network. The k-th element of the prediction is

$$F_{\mathrm{MM}}(x)_k = P(y = k|z_\theta(x)), \tag{5}$$

where $k \in [L]$. In the training phase, the loss function[3] for the MM-LDA network is $\mathcal{L}_{\mathrm{MM}}(x, y) = -1_y^\top \log F_{\mathrm{MM}}(x)$, which is the cross-entropy between the one-hot true label 1_y and the prediction $F_{\mathrm{MM}}(x)$. By minimizing the loss function w.r.t θ on the training set $\mathcal{D} := \{(x_i, y_i)\}_{i \in [N]}$, we can obtain the optimal parameters θ_{MM} for the MM-LDA network as $\theta_{\mathrm{MM}} = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\mathrm{MM}}(x_i, y_i)$. In Alg. 2 we demonstrate the complete training phase. In the test phase, the MM-LDA network returns the predicted label $\hat{y}_{\mathrm{MM}} = \arg\max_k F_{\mathrm{MM}}(x)_k$, where the set of means μ^* is the same as the one used in training.

Fig. 1 provides an intuitive comparison between MM-LDA networks and SR networks (See Sec. 4.2 for details). For SR networks, though the latent features are discriminative, the distribution is not as orderly as that for MM-LDA networks.

[3] More discussion on the training loss function for the MM-LDA network can be found in Appendix B.1



(a) Resnet-32 (SR)

(b) Resnet-32 (MM-LDA)

Figure 1. t-SNE visualization of the latent features on CIFAR-10. The index numbers indicate classes 0 to 9, where each number locates on the median position of the corresponding vectors.

This is because SR networks do not explicitly model the distribution of z, while MM-LDA does model it as a well-structured MMD. This structure can influence the nonlinear transformation network via back-propagation as in Alg. 2.

In addition to better robustness in the adversarial setting, the MM-LDA network should also perform better than the SR network on the input with a class-biased distribution, i.e., a distribution with different class priors. This can be intuitively illustrated by the conclusions in Section 2.3 that a larger value of $|\zeta|$ implies lower relative efficiency of LR to LDA. Since ζ denotes the log-ratio of class-priors, a larger value of $|\zeta|$ indicates more biased class priors, i.e., bigger gaps among class priors. Thus in the multi-class cases, acting on the input with biased class priors should intuitively result in low relative efficiency of SR to LDA, and further to MM-LDA.

4. Experiments

We now experimentally demonstrate that the MM-LDA networks are more robust in the adversarial setting while maintaining state-of-the-art performance on normal examples, and have better performance on class-biased datasets.

4.1. Setup

We choose the widely used MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009) datasets. MNIST consists of grey images of handwritten digits in classes 0 to 9, and CIFAR-10 consists of color images in 10 different classes. Each dataset has 60,000 images, of which 50,000 are in the training set and the rest are in the test set. The pixel values of images in both sets are scaled to the interval $[-0.5, 0.5]$ before fed into classifiers. The baseline is the most common SR network (Goodfellow et al., 2016). The empirical class prior $\hat{\pi}_k$ of a dataset with N samples is $\pi_k = N_k/N$, where N_k is the number of samples in class k. Then a dataset is class-unbiased if $\forall k \in [L], \hat{\pi}_k = 1/L$ for both training and testing sets; otherwise class-biased.

4.2. Performance on Normal Examples

We first test the performance on normal examples (i.e., the samples in the original datasets without any perturbations). We implement Resnet-32 (He et al., 2016a) on MNIST and CIFAR-10, which uses the SR model as the linear classifier. This network will be denoted by Resnet-32 (SR). Our MM-LDA network shares the same architecture of nonlinear transformation as Resnet-32 (SR) while uses the MM-LDA procedure for classification, and we denote it by Resnet-32 (MM-LDA). The number of training steps is 20,000 on MNIST and 90,000 on CIFAR-10 for both networks. Here we apply the training setting introduced in He et al. (2016b) to train the Resnet-32 (SR). To train the Resnet-32 (MM-LDA), we simply use the same training setting as the Resnet-32 (SR), except that we apply the adaptive optimization method—Adam (Kingma & Ba, 2015) rather than the momentum SGD used in He et al. (2016b), to avoid extra effort on tuning training hyperparameters.[4]

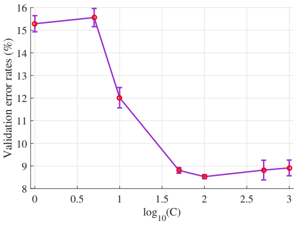


Figure 2. Validation error rates (%) of the MM-LDA networks w.r.t different values of $\log_{10}(C)$ on CIFAR-10.

When applying the MM-LDA network, the only hyperparameter is the square norm C of the Gaussian means in MMD. If C is too small, the conditional Gaussian components in MMD will largely overlap to each other, which makes the optimal Bayes error rate be high, and further results in a high error rate for the LDA classifier. Besides, if C is too large, the magnitudes of the transformation parameters θ will also tend to sharply increase during the training procedure, which makes the MM-LDA network easy to overfit. In our experiments, we empirically choose the value of C by doing 5-fold cross-validation on the training set. Fig. 2 shows the validation error rates of the MM-LDA networks w.r.t $\log_{10}(C)$ on CIFAR-10. We find that when $\log_{10}(C) = 2$, i.e., $C = 100$ the MM-LDA network has the lowest average validation error rate with a small value of standard deviation. Thus hereafter we will set $C = 100$.

Table 2 shows the classification error rates on the test sets of both datasets. We can see that the MM-LDA network maintains state-of-the-art performance on normal examples. Furthermore, in Fig. 1 we apply the t-SNE technique (Maaten & Hinton, 2008) to visualize the latent feature vectors on 1,000 randomly sampled test images of CIFAR-10. We can find that the trained MM-LDA network maps the data distribution in the input space to a much more regular distribution in the latent feature space, as stated before. Note that class 3 and class 5 are close to each other in Fig. 1b, which is reasonable since they are separately 'cat' and 'dog', even a human observer will sometimes misidentify them.

[4] We also try to substitute Adam for the momentum SGD in the training phase of Resnet-32 (SR), and we found that the momentum SGD makes Resnet-32 (SR) have lower error rates on both datasets.

Table 1. Classification accuracy (%) on adversarial examples of MNIST and CIFAR-10. The investigated values of perturbation are 0.04, 0.12, and 0.20. **Boldface** indicates the best result under certain combination of a value of perturbation and an attacking method.

Perturbation	Model	MNIST				CIFAR-10			
		FGSM	BIM	ILCM	JSMA	FGSM	BIM	ILCM	JSMA
0.04	Resnet-32 (SR)	93.6	87.9	94.8	92.9	20.0	5.5	0.2	65.6
	Resnet-32 (SR) + SAT	86.7	68.5	98.4	-	24.4	7.0	0.4	-
	Resnet-32 (SR) + HAT	88.7	96.3	**99.8**	-	30.3	5.3	1.3	-
	Resnet-32 (MM-LDA)	**99.2**	**99.2**	99.0	**99.1**	**91.3**	**91.2**	**70.0**	**91.2**
0.12	Resnet-32 (SR)	28.1	3.4	20.9	56.0	10.2	4.1	0.3	20.5
	Resnet-32 (SR) + SAT	40.5	8.7	88.8	-	88.2	6.9	0.1	-
	Resnet-32 (SR) + HAT	40.3	40.1	92.6	-	44.1	8.7	0.0	-
	Resnet-32 (MM-LDA)	**99.3**	**98.6**	**99.6**	**99.7**	**90.7**	**90.1**	**42.5**	**91.1**
0.20	Resnet-32 (SR)	15.5	0.3	1.7	25.6	10.7	4.2	0.6	11.5
	Resnet-32 (SR) + SAT	17.3	1.1	69.4	-	**91.7**	9.4	0.0	-
	Resnet-32 (SR) + HAT	10.1	10.5	46.1	-	40.7	6.0	0.2	-
	Resnet-32 (MM-LDA)	**97.5**	**97.3**	**96.6**	**99.6**	89.5	**89.7**	**31.2**	**91.8**

Table 2. Error rates (%) on the test sets of MNIST and CIFAR-10.

Model	MNIST	CIFAR-10
Resnet-32 (SR)	0.38	**7.13**
Resnet-32 (MM-LDA)	**0.35**	8.04

Table 3. Average minimal distortions of the adversarial examples crafted by the C&W attack on MNIST and CIFAR-10.

Model	MNIST	CIFAR-10
Resnet-32 (SR)	8.56	0.67
Resnet-32 (MM-LDA)	**16.32**	**2.80**

4.3. Performance in the Adversarial Setting

Now we test the robustness of MM-LDA networks in the adversarial setting. Adversarial training is one of the most common and effective methods to improve the robustness of classifiers on iterative-based attacks (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017b; Tramèr et al., 2017). Thus in addition to the SR networks trained on normal examples, we also treat the SR networks enhanced by adversarial training as stronger baselines. We construct the enhanced baselines by first crafting adversarial examples on the trained SR networks, then fine-tuning the networks on the mixture of the normal examples and crafted adversarial examples. More technical details are in Appendix B.2.

For more complete analysis, we apply two kinds of adversarial training methods to enhance baselines. They differ in the choices of adversarial examples to fine-tune the classifiers:

Specific Adversarial Training (SAT) fine-tunes the classifiers on the adversarial examples crafted by the same attack with the same value of perturbation ϵ as that when attacking the classifiers. Similar strategy is used in (Szegedy et al., 2014; Goodfellow et al., 2015).

Hybrid Adversarial Training (HAT) fine-tunes the classifiers on the adversarial examples crafted by the same attack as that when attacking the classifiers, but with various values of ϵ. Specifically, we uniformly choose ϵ from the interval $[0.02, 0.20]$ when crafting adversarial examples for HAT. Similar strategy is used in (Kurakin et al., 2017b).

Table 1 presents the classification accuracy of the networks on the adversarial examples crafted by iterative-based attacks. We investigate on three different values of perturbation ϵ: 0.04, 0.12 and 0.20 (See Section 2.2 for ϵ). Usually an adversarial noise with perturbation larger than 0.05 is perceivable by human eyes. From the results, we can see that both SAT and HAT can effectively enhance the original baselines to stronger ones. However, the adversarial training methods require extra computational cost and are less effective on multi-step methods, e.g., BIM and ILCM (Kurakin et al., 2017b). By contrast, the MM-LDA network significantly improves the robustness on iterative-based attacks compared to almost all the baselines. This is because in the MM-LDA networks, normal examples distribute as a MMD in the latent feature space, which makes it more difficult for adversaries to move a normal example from its original class to other classes. Note that we do not adversarially fine-tune Resnet-32 (SR) on the JSMA attack, since it is computationally expensive to craft an adversarial example by JSMA, which makes adversarial training inefficient.

We also apply the optimization-based C&W attack on the trained networks. Since there is yet no method including adversarial training to effectively defend the C&W attack under the white-box threat model (Carlini & Wagner, 2017a;b), we only compare between Resnet-32 (SR) and Resnet-32 (MM-LDA). In the C&W attack, we set the binary search steps for the constant c be 9, and the maximal number of iteration steps for each value of c be 10,000. This setting is strong enough, so that the crafted adversarial examples can evade both the SR and MM-LDA networks with nearly 100% success rate. As shown in Table 3, the average minimal distortions of adversarial examples on the MM-LDA networks are much larger than those on the SR networks. Here the distortion is defined in Szegedy et al. (2014), where the pixel values of images are in the interval $[0, 255]$ when calculating it. This results mean that the C&W attack has to add much larger noises to successfully evade the MM-LDA networks, as theoretically demonstrated in Sec. 3.1.

Furthermore, we find that when applying the C&W attack

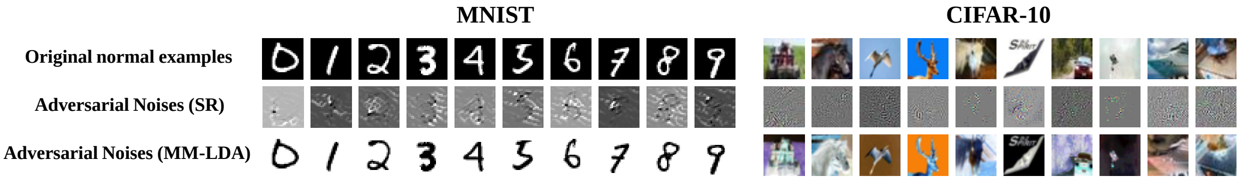


Figure 3. Some normal examples with the semantic adversarial noises crafted by the C&W attack on MNIST and CIFAR-10. The shown noises crafted for the MM-LDA networks are much more semantic, while similar meaningful noises are hardly observed in those crafted for the SR networks. *However, most of the adversarial noises crafted for the MM-LDA networks still seem like random noises.*

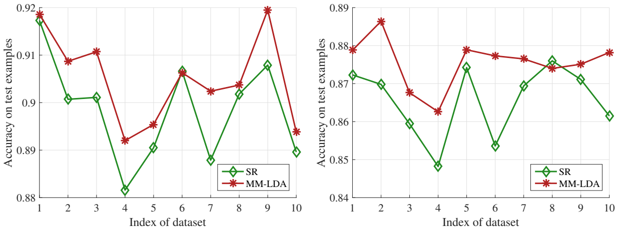
on the MM-LDA networks, some of the adversarial noises have the same semantic meanings as their corresponding normal examples (around $1\% \sim 5\%$ of all noises), while similar phenomenon can hardly be observed when attacking on the SR networks (less than 0.1% of all noise). We show some of the semantic noises in Fig. 3. The adversarial noise is calculated as $(x^* - x)/2$ to keep the pixel values of the noise in the interval $[-0.5, 0.5]$. This result indicates that MM-LDA networks can learn more robust features, such that on the shown normal examples, the optimal attacking strategy that the C&W attack finds for MM-LDA networks is to weaken the features of the normal examples as a whole, rather than adding meaningless noise as for the SR networks.

4.4. Performance on Class-biased Datasets

Finally, we evaluate on class-biased datasets, which are more realistic though many sets were artificially constructed as class-unbiased, e.g., CIFAR-10. We construct the class-biased datasets by randomly sampling each data point of class i from CIFAR-10 with probability $\alpha_i, i \in L$, where $L = 10$ for CIFAR-10. Specifically, let $\alpha = (\alpha_0, \cdots, \alpha_9)$, $\mathcal{D}$ be the training or test set of CIFAR-10, then the constructed class-biased dataset is $\mathcal{D}_\alpha^{\text{bias}}$ that $\forall (x, y) \in \mathcal{D}$, $P((x, y) \in \mathcal{D}_\alpha^{\text{bias}}) = \alpha_y$. Then the empirical class priors of $\mathcal{D}_\alpha^{\text{bias}}$ have expectations as $\mathbb{E}[\hat{\pi}_k] = \alpha_k / \|\alpha\|_1$. We choose two typical kinds of bias probability α as below:

Bias Probability 1 (BP1) has $\alpha = (0.1, 0.2, 0.3, \cdots, 1.0)$. To avoid the system error caused by certain permutation of the elements in α, we randomly rearrange the elements in α to get 10 counterparts $\alpha^{(0)}, \cdots \alpha^{(9)}$. The publicly available datasets with similar class-prior distributions as BP1 including the IMDB-WIKI dataset (Rothe et al., 2015) for age and gender prediction, and the KITTI dataset (Geiger et al., 2012) for autonomous driving.

Bias Probability 2 (BP2) has $\alpha = (0.2, \cdots, 0.2, 1.0)$. Since there is only one element in α that equals to 1.0 and the others all equal to 0.2, we assign 1.0 in turn to 10 different classes to avoid the system error, and similarly get 10 counterparts $\alpha^{(0)}, \cdots, \alpha^{(9)}$. The Caltech101 dataset (Fei-Fei et al., 2007) and the large-scale ImageNet dataset (Deng et al., 2009) have similar class-prior distributions as BP2.



(a) BP1

(b) BP2

Figure 4. Classification accuracy on the test sets of class-biased datasets. Each index of dataset corresponds to a counterpart of the bias probability. The original class-unbiased dataset is CIFAR-10.

We separately apply the counterparts of BP1 and BP2 on both the training and test sets of CIFAR-10 to construct totally 20 class-biased datasets, i.e., both the training and test sets of each constructed dataset are class-biased. Then we train Resnet-32 (SR) and Resnet-32 (MM-LDA) on each class-biased dataset. Fig. 4 shows the test accuracy of the trained networks on the 20 class-biased datasets. Note that when training and testing the MM-LDA networks, we set the class priors as uniform $\pi_k = 1/L$ to calculate the prediction $F_{\text{MM}}(x)$ rather than setting $\pi_k = \alpha_k / \|\alpha\|_1$. By doing this we give a fair comparison between the SR and the MM-LDA networks, since SR is a discriminant model that cannot exploit the information of class priors. We can see that the MM-LDA networks still perform better than the SR networks on almost all the datasets constructed by the counterparts of BP1 and BP2. This result indicates that the better performance of the MM-LDA networks on class-biased datasets comes from the intrinsic superiority of the MM-LDA networks, not from the extra knowledge on class priors. We also try to set $\pi_k = \alpha_k / \|\alpha\|_1$, and find that the difference of $F_{\text{MM}}(x)$ between the two settings is small, which most likely leads to the same predicted label. This is because when the class priors are not too biased, i.e., the value of $\max_{i,j \in [L]} |\log(\pi_i / \pi_j)|$ is not too large, the exponential terms in Eq. (5) will dominate the calculation of $F_{\text{MM}}(x)$ since we choose a relatively large $C = 100$.

5. Conclusions

In this paper we propose the novel MM-LDA network. The MM-LDA network is much more robust in the adversarial setting with theoretical guarantees, while maintaining state-of-the-art performance on normal examples. The MM-LDA network also performs better on class-biased datasets. Our network is easy to implement and can be naturally combined with different nonlinear transformation architectures and training methods designed for the SR network.

Acknowledgements

This work was supported by NSFC Projects (Nos. 61620106010, 61621136008, 61332007), Beijing NSF Project (No. L172037), Tiangong Institute for Intelligent Computing, NVIDIA NVAIL Program, Siemens and Intel.

References

Carlini, Nicholas and Wagner, David. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*, 2017a.

Carlini, Nicholas and Wagner, David. Adversarial examples are not easily detected: Bypassing ten detection methods. *ACM Workshop on Artificial Intelligence and Security*, 2017b.

Coates, Adam, Ng, Andrew, and Lee, Honglak. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

Efron, Bradley. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352):892–898, 1975.

Fallah, Faezeh, Tsanev, Doychin Mariyanov, Yang, Bin, Walter, Sven, and Bamberg, Fabian. A novel objective function based on a generalized kelly criterion for deep learning. In *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2017*, pp. 84–89. IEEE, 2017.

Fei-Fei, Li, Fergus, Rob, and Perona, Pietro. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Geiger, Andreas, Lenz, Philip, and Urtasun, Raquel. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, 2016.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *The International Conference on Learning Representations (ICLR)*, 2015.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016a.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pp. 630–645. Springer, 2016b.

Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Huang, Chen, Loy, Chen Change, and Tang, Xiaoou. Discriminative sparse neighbor approximation for imbalanced learning. *IEEE transactions on neural networks and learning systems*, 2017.

Huang, Fu Jie and LeCun, Yann. Large-scale learning with svm and convolutional for generic object categorization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 284–291. IEEE, 2006.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *The International Conference on Learning Representations (ICLR)*, 2015.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. Technical report, 2009.

Kurakin, Alexey, Goodfellow, Ian, and Bengio, Samy. Adversarial examples in the physical world. *The International Conference on Learning Representations (ICLR) Workshops*, 2017a.

Kurakin, Alexey, Goodfellow, Ian, and Bengio, Samy. Adversarial machine learning at scale. *The International Conference on Learning Representations (ICLR)*, 2017b.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Liu, Weiyang, Wen, Yandong, Yu, Zhiding, and Yang, Meng. Large-margin softmax loss for convolutional neural networks. In *ICML*, pp. 507–516, 2016.

Maaten, Laurens van der and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

Moosavi-Dezfooli, Seyed-Mohsen, Fawzi, Alhussein, and Frossard, Pascal. Deepfool: a simple and accurate method to fool deep neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

Ngiam, Jiquan, Chen, Zhenghao, Chia, Daniel, Koh, Pang W, Le, Quoc V, and Ng, Andrew Y. Tiled convolutional neural networks. In *Advances in neural information processing systems*, pp. 1279–1287, 2010.

Nguyen, Anh, Yosinski, Jason, and Clune, Jeff. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 427–436, 2015.

Pang, Tianyu, Du, Chao, Dong, Yinpeng, and Zhu, Jun. Towards robust detection of adversarial examples. *arXiv preprint arXiv:1706.00633*, 2017.

Papernot, Nicolas, McDaniel, Patrick, Jha, Somesh, Fredrikson, Matt, Celik, Z Berkay, and Swami, Ananthram. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387. IEEE, 2016.

Rothe, Rasmus, Timofte, Radu, and Van Gool, Luc. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 10–15, 2015.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. *The International Conference on Learning Representations (ICLR)*, 2014.

Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, and Wojna, Zbigniew. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Tang, Yichuan. Deep learning using linear support vector machines. *International Conference on Machine Learning (ICML) Workshops*, 2013.

Tramèr, Florian, Kurakin, Alexey, Papernot, Nicolas, Boneh, Dan, and McDaniel, Patrick. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Appendix

A. Proof

Theorem 1. *The relative efficiency of logistic regression to linear discriminant analysis is*

$$\mathrm{Eff}_p(\zeta, \Delta) = (Q_1 + (p-1)Q_2)/(Q_3 + (p-1)Q_4),$$

where $Q_2 = 1 + \pi_0\pi_1\Delta^2$, $Q_4 = \frac{1}{A_0}$ *and*

$$\begin{aligned} Q_1 &= \begin{pmatrix}1 & \frac{\zeta}{\Delta}\end{pmatrix} \begin{bmatrix} 1+\frac{\Delta^2}{4} & (\pi_0-\pi_1)\frac{\Delta}{2} \\ (\pi_0-\pi_1)\frac{\Delta}{2} & 1+2\pi_0\pi_1\Delta^2 \end{bmatrix} \begin{bmatrix} 1 \\ \frac{\zeta}{\Delta} \end{bmatrix}, \\ Q_3 &= \begin{pmatrix}1 & \frac{\zeta}{\Delta}\end{pmatrix} \frac{1}{A_0A_2 - A_1^2} \begin{bmatrix} A_2 & A_1 \\ A_1 & A_0 \end{bmatrix} \begin{bmatrix} 1 \\ \frac{\zeta}{\Delta} \end{bmatrix}. \end{aligned}$$

Proof. The proof can be found in Efron (1975). □

Theorem 2. *If* $\pi_i = \pi_j$*, the expectation of the distance* $d_{(i,j)}$ *is a function of the Mahalanobis distance* $\Delta_{i,j}$*:*

$$\mathbb{E}[d_{(i,j)}] = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\Delta_{i,j}^2}{8}\right) + \frac{1}{2}\Delta_{i,j}\left[1 - 2\Phi(-\frac{\Delta_{i,j}}{2})\right],$$

where $\Phi(\cdot)$ *is the normal cumulative distribution function.*

Proof. Since $d_{(i,j)}$ is the distance of $x_{(i)}$ to the decision boundary between class i and j decided by the Fisher's linear discriminant function $\lambda_{i,j}(x) = \beta_{i,j} + \alpha_{i,j}^\top x = 0$, where

$$\begin{aligned} \beta_{i,j} &= \log(\pi_i/\pi_j) + \frac{1}{2}(\|\mu_j\|_2^2 - \|\mu_i\|_2^2), \\ \alpha_{i,j}^\top &= (\mu_i - \mu_j)^\top. \end{aligned}$$

We have

$$d_{(i,j)} = \frac{|\beta_{i,j} + \alpha_{i,j}^\top x_{(i)}|}{\|\alpha_{i,j}\|_2} = \frac{|\beta_{i,j} + \alpha_{i,j}^\top x_{(i)}|}{\Delta_{i,j}}.$$

Because $x_{(i)}$ is sampled from the conditional Gaussian distribution of class i, there is

$$x_{(i)} \sim \mathcal{N}(\mu_i, I).$$

Let $H_{i,j} = \beta_{i,j} + \alpha_{i,j}^\top x_{(i)}$, and $\zeta_{i,j} = \log(\pi_i/\pi_j)$, then according to the property of Gaussian distribution we can know that

$$H_{i,j} \sim \mathcal{N}(\mu'_{i,j}, \sigma_{i,j}^2),$$

where

$$\begin{aligned} \mu'_{i,j} &= \beta_{i,j} + \alpha_{i,j}^\top \mu_i \\ &= \zeta_{i,j} + \frac{1}{2}(\|\mu_j\|_2^2 - \|\mu_i\|_2^2) + (\mu_i - \mu_j)^\top \mu_i \\ &= \zeta_{i,j} + \frac{1}{2}\|\mu_i - \mu_j\|_2^2 \\ &= \zeta_{i,j} + \frac{1}{2}\Delta_{i,j}^2, \end{aligned}$$

and $\sigma_{i,j}^2 = \alpha_{i,j}^\top I \alpha_{i,j} = \Delta_{i,j}^2$. Thus $|H_{i,j}|$ distributes as a *Folded Gaussian (Normal) Distribution*. From the property of folded Gaussian distribution we know that

$$\mathbb{E}[|H_{i,j}|] = \sqrt{\frac{2}{\pi}}\sigma_{i,j} \exp(-\frac{{\mu'_{i,j}}^2}{2{\sigma_{i,j}}^2}) + \mu'_{i,j}[1 - 2\Phi(-\frac{\mu'_{i,j}}{\sigma_{i,j}})].$$

For notation clarity, we let

$$\alpha_{i,j} = \frac{\mu'_{i,j}}{\sigma_{i,j}} = \frac{1}{2}\Delta_{i,j} + \zeta_{i,j}/\Delta_{i,j}.$$

Since $\mathbb{E}[d_{(i,j)}] = \mathbb{E}[|H_{i,j}|]/\Delta_{i,j}$, we have

$$\mathbb{E}[d_{(i,j)}] = \sqrt{\frac{2}{\pi}} \exp(-\frac{\alpha_{i,j}^2}{2}) + \alpha_{i,j}[1 - 2\Phi(-\alpha_{i,j})].$$

The derivative of $\mathbb{E}[d_{(i,j)}]$ to $\Delta_{i,j}$ is

$$\begin{aligned} \frac{\partial \mathbb{E}[d_{(i,j)}]}{\partial \Delta_{i,j}} &= \frac{\partial \mathbb{E}[d_{(i,j)}]}{\partial \alpha_{i,j}} \cdot \frac{\partial \alpha_{i,j}}{\partial \Delta_{i,j}} \\ &= [1 - 2\Phi(-\frac{1}{2}\Delta_{i,j} - \frac{\zeta_{i,j}}{\Delta_{i,j}})](\frac{1}{2} - \frac{\zeta_{i,j}}{\Delta_{i,j}^2}) \\ &= [1 - 2\Phi(-\Delta_{i,j}(\frac{1}{2} - \frac{\zeta_{i,j}}{\Delta_{i,j}^2}))](\frac{1}{2} - \frac{\zeta_{i,j}}{\Delta_{i,j}^2}) \\ &\geq 0, \end{aligned}$$

where the Mahalanobis distance $\Delta_{i,j}$ is non-negative. Specially, when $\pi_i = \pi_j$, i.e., $\zeta_{i,j} = 0$, there is

$$\mathbb{E}[d_{(i,j)}] = \sqrt{\frac{2}{\pi}} \exp(-\frac{\Delta_{i,j}^2}{8}) + \frac{1}{2}\Delta_{i,j}[1 - 2\Phi(-\frac{\Delta_{i,j}}{2})].$$

□

Theorem 3. *Assume that* $\sum_{i=1}^{L} \mu_i = 0$ *and* $\|\mu\|_2^2 = C$. *Then there has*

$$\overline{\mathrm{RB}} \leq \sqrt{\frac{LC}{2(L-1)}}.$$

The equality holds if and only if

$$\mu_i^\top \mu_j = \begin{cases} C, & i = j, \\ C/(1-L), & i \neq j, \end{cases}$$

where $i, j \in [L]$ *and* $\mu_i, \mu_j \in \mu$.

Proof. According to the definition of $\overline{\mathrm{RB}}$, we have

$$\begin{aligned} \overline{\mathrm{RB}} &= \frac{1}{2} \min_{i,j \in [L]} \Delta_{i,j} \\ &= \frac{1}{2} \sqrt{\min_{i,j \in [L]} \Delta_{i,j}^2} \\ &\leq \frac{1}{2} \sqrt{\frac{1}{L(L-1)} \sum_{i \neq j} \Delta_{i,j}^2} \\ &= \frac{1}{2} \sqrt{\frac{1}{L(L-1)} \sum_{i \neq j} (\|\mu_i\|_2^2 + \|\mu_j\|_2^2 - 2\mu_i^\top \mu_j)} \\ &= \frac{1}{2} \sqrt{\frac{2}{L} \sum_{i \in [L]} \|\mu_i\|_2^2 - \frac{1}{L(L-1)} \sum_{i \neq j} 2\mu_i^\top \mu_j} \\ &= \frac{1}{2} \sqrt{\frac{2}{L-1} \sum_{i \in [L]} \|\mu_i\|_2^2 - \frac{1}{L(L-1)} (\sum_{i \in [L]} \mu_i)^2}, \end{aligned}$$

Since $\sum_{i=1}^{L} \mu_i = 0$ and $\|\mu\|_2^2 = C$, we further have

$$\begin{aligned} \overline{\mathrm{RB}} &\leq \sqrt{\frac{1}{2(L-1)} \sum_{i \in [L]} \|\mu_i\|_2^2} \\ &\leq \sqrt{\frac{LC}{2(L-1)}}. \end{aligned}$$

Note that the final equality holds if and only if all the equalities hold, i.e., there are

$$\|\mu_i\|_2^2 = C, \forall i \in [L],$$

and

$$\Delta_{i,j} = \text{constant}, \forall i \neq j.$$

Thus we can easily derive that the final equality holds if and only if

$$\mu_i^\top \mu_j = \begin{cases} C, & i = j, \\ C/(1-L), & i \neq j, \end{cases}$$

where $i, j \in [L]$ and $\mu_i, \mu_j \in \mu$.

□

B. More Discussions and Details

In this section we discuss more on the loss function of the MM-LDA network, and the choice of the square norm of C of MMD. Besides, we provide technical details of the adversarial training methods we use in our experiments.

B.1. The Loss Function of the MM-LDA Network

Considering that the network with parameters θ induces a joint distribution on the latent feature z and the label y as $Q_\theta(z, y)$. We denote the MMD as $P(z, y)$, $\mathbb{H}(P, Q)$ as the cross-entropy for the distributions P and Q. Then the training objective could be designed as

$$\begin{aligned} \mathbb{H}(Q_\theta, P) &= \mathbb{E}_{(z,y)\sim Q_\theta}[-\log P(y|z) - \log P(z)] \\ &= \mathbb{E}_{(z,y)\sim Q_\theta}[-\log P(y|z)] + \mathbb{E}_{z \sim Q_\theta'}[-\log P(z)]. \end{aligned}$$

Here $Q_\theta^{'}$ is the marginal distribution of Q_θ for z. Since we are focusing on classification tasks, we assume for tractability that the marginal distribution $Q_\theta^{'}(z)$ is consistent with it of the MMD, i.e., $P(z)$. Therefore, minimizing $\mathbb{H}(Q_\theta, P)$ equals to minimizing $\mathbb{E}_{(z,y)\sim Q_\theta}[-\log P(y|z)]$, which further leads to the loss function $\mathcal{L}_{\mathrm{MM}}$ under the Monte Carlo approximation. In practice, the gap between $Q_\theta^{'}(z)$ and $P(z)$ would not influence the performance, as shown in our experiment results.

In order to better gather the latent feature vectors to their corresponding conditional Gaussian distributions, the loss function of the MM-LDA network should be

$$\mathcal{L}_{\mathrm{MM}} = -F_{\mathrm{MM}}(\mu_y^*)^\top \log F_{\mathrm{MM}}(x),$$

where $F_{\mathrm{MM}}(\mu_y^*)_k = P(y = k|z = \mu_y^*), k \in [L]$. $F_{\mathrm{MM}}(\mu_y^*)$ is the prediction vector on the mean vector μ_y^* in MMD. We further have

$$
\begin{aligned}
F_{\mathrm{MM}}(\mu_y^*)_y &= \frac{\exp(2{\mu_y^*}^\top \mu_y^*)}{\sum_{i\in[L]}\exp(2{\mu_y^*}^\top \mu_i^*)} \\
&= \frac{1}{1+\sum_{i\neq y}\exp[2{\mu_y^*}^\top(\mu_i^*-\mu_y^*)]} \\
&= \frac{1}{1+(L-1)\exp(-\frac{2LC}{L-1})}.
\end{aligned}
$$

Thus the L_∞-distance between the one-hot label vector 1_y and $F_{\mathrm{MM}}(\mu_y^*)$ is

$$
\begin{aligned}
\|1_y - F_{\mathrm{MM}}(\mu_y^*)\|_\infty &\leq \left|1-F_{\mathrm{MM}}(\mu_y^*)_y\right| \\
&= \frac{1}{1+\frac{1}{L-1}\exp(\frac{2LC}{L-1})}.
\end{aligned}
$$

It is easy to see that the gap between 1_y and $F_{\mathrm{MM}}(\mu_y^*)$ rapidly decreases w.r.t C. For instance, when $C > 10$, $L = 10$, we numerically have $\|1_y - F_{\mathrm{MM}}(\mu_y^*)\|_\infty \leq 10^{-8}$. Therefore we use the one-hot label 1_y in the loss function because of its simplicity.

B.2. Technical Details of Adversarial Training

Our experiments are done on NVIDIA Tesla P100 GPUs. The number of the adversarial fine-tuning steps on MNIST is 10,000 and on CIFAR-10 is 30,000. We apply a constant learning rate of 0.01 on both datasets. The mixing ratio of normal examples and adversarial examples is 1:1. Averagely the time cost to craft an adversarial example using FGSM, BIM and ILCM is less than 0.1 seconds, while using JSMA is around 5 seconds. This makes adversarial training on JSMA computationally expensive.

References

Efron, Bradley. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352):892–898, 1975.