# Research Paper - Presentation

Dontha Aarthi - CS20BTECH11015

June 25, 2021

# Title and Authors

## Title

Deep Minimax Probability Machine

## Authors

1. Lirong He, SMILE Lab, School of Computer Science and Engineering University of Electronic Science and Technology of China Chengdu, China, lirong he@std.uestc.edu.cn
2. Ziyi Guo, Cloud and Smart Industries Group Tencent Guangzhou, China, ziyiguo94@gmail.com
3. Kaizhu Huang, Department of EEE, Xi'an Jiaotong-Liverpool University, Suzhou, China, Kaizhu.Huang@xjtlu.edu.cn
4. Zenglin Xu, SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu, China, Center for Artificial Intelligence, Peng Cheng Laboratory Shenzhen, China, zenglin@gmail.com

# Index Terms

1. **Deep Neural Networks (DNNs)**: Neural networks are layers of nodes. Each mathematical manipulation as such is considered as a layer. Nodes within each layer are connected to adjacent layers.If there are more hidden layers, its called Deep Neural Network.
2. **Adversarial attacks**:An adversarial attack consists of subtly modifying an original image in such a way that the changes are almost undetectable to the human eye. The modified image is called an adversarial image, and when submitted to a classifier is misclassified, while the original one is correctly classified.
3. **Minimax Probability Machine (MPM)**: It is the method used for maximising the minimum probability of a regression model for all possible distributions with known mean and covariance matrix.

# Abstract

1. DNNs are adept at learning effective representation and have demonstrated significant success in a wide variety of applications, such as image classification, speech recognition and language translation. However, recent advances show that they are vulnerable to adversarial examples.
2. Although the adversarial example is only slightly different from the input sample, the neural network classifies it as the wrong class.
3. In order to alleviate this problem, we propose the Deep Minimax Probability Machine (DeepMPM), which applies MPM to deep neural networks in an end-to-end fashion.
4. DeepMPM could take advantage of global information by introducing the global statistics of data, i.e., the mean and covariance of data, control misclassification probabilities robustly in the worst case for future data, and do well in learning effective hidden representation.

# Deep Minimax Probability Machine

1. Combining MPM with DNNs could inherit the good advantages of both MPM and DNNs where robustness and accuracy would be integrated.
2. MPM can obtain the upper bound on the probability of misclassification for future data, i.e., the worst-case accuracy and hence it leads to a robust classifier.
3. We put the MPM at the top of a deep neural network, as shown in Figure below.
4. Specifically, instead of maximizing the likelihood of labels for data, we employ the objective function of MPM to promote our model to take into account global information.

# Graphical Illustration of Deep MPM

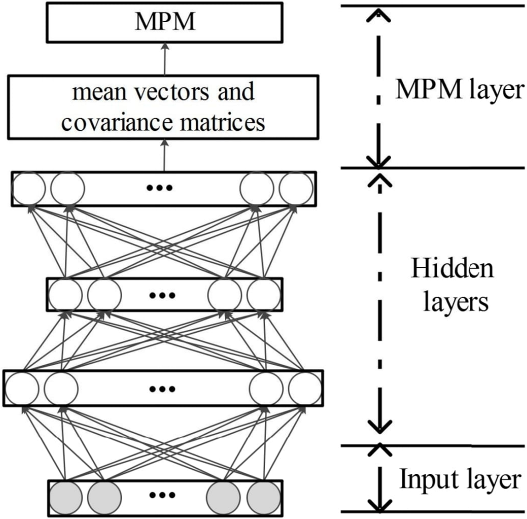

Figure: Deep MPM model

# Minimax probability machine

1. Minimax Probability Machine (MPM) tries to minimize the upper bound of the probability of misclassification of future data in a worst-case setting.
2. No assumptions with respect to the data distribution are required in MPM, while those assumptions lack generality and are often invalid.

# Minimax probability machine

Let x and y denote random vectors with mean vectors and covariance matrices given by x $\sim (\overline{x}, \sum_x)$ and y $\sim (\overline{y}, \sum_y)$ respectively in a binary classification problem, where x, $\overline{x}$, y, $\overline{y} \in \mathbb{R}^n$ and $\sum_x, \sum_y \in \mathbb{R}^{n\times n}$. MPM seeks to determine the hyperplane $a^\top z = b$ ($a, z \in \mathbb{R}^n$ and $b \in \mathbb{R}$) which separates the two classes of data with maximal probability. The form of the MPM model is as follows:

$$\begin{aligned} \max\nolimits_{\alpha,a,b}\ \alpha \text{ s.t. } & \inf\ \Pr\left(a^\top x \geq b\right) \geq \alpha, \\ & \underline{\inf\ \Pr\left(a^\top y \leq b\right) \geq \alpha.} \end{aligned} \tag{1}$$

Where $\alpha$ denotes the worst case accuracy of the future data.

# Minimax probability machine(contd.)

## Lemma

*Given* $a \neq 0$ *and* $b$*, such that* $a^\top z \leq b$ *and* $\beta \in [0,1)$*, the condition*

$$\underline{inf_{\,y \sim (\overline{y}, \sum_y)} \Pr\left(a^\top y \leq b\right) \geq \beta,}$$

*holds if and only if* $b - a^\top \overline{y} \geq \kappa(\beta)\sqrt{a^\top \sum_y a}$ *with* $\kappa(\beta) = \sqrt{\frac{\beta}{1-\beta}}$.

By using the above lemma, (1) can be written as

$$\max_{\alpha, a, b} \alpha \text{ s.t.}$$

$$-b + a^\top \overline{x} \geq \kappa(\alpha)\sqrt{a^\top \sum_x a} \qquad (2)$$

$$b - a^\top \overline{y} \geq \kappa(\alpha)\sqrt{a^\top \sum_y a} \qquad (3)$$

# Minimax probability machine(contd.)

Where $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$.
From (2) and (3), we get

$$a^\top \overline{y} + \kappa(\alpha)\sqrt{a^\top \sum_y a} \leq b \leq a^\top \overline{x} - \kappa(\alpha)\sqrt{a^\top \sum_x a} \tag{4}$$

On eliminating b from (4), we get

$$a^\top (\overline{x} - \overline{y}) \geq \kappa(\alpha)\left(\sqrt{a^\top \sum_x a} + \sqrt{a^\top \sum_y a}\right) \tag{5}$$

# Minimax probability machine(contd.)

Without loss of generality, we can set $a^\top(\overline{x}-\overline{y})=1$. Thus (5) changes to:

$$\max_{\alpha,a,b}\ \alpha \text{ s.t.}$$

$$1 \geq \kappa(\alpha)\left(\sqrt{a^\top \textstyle\sum_x a}+\sqrt{a^\top \textstyle\sum_y a}\right), \tag{6}$$

$$a^\top(\overline{x}-\overline{y})=1$$

Since, $\kappa(\alpha)=\sqrt{\frac{\alpha}{1-\alpha}}$, it increases monotonically with $\alpha$, so maximising $\alpha$ is equivalent to maximising $\kappa(\alpha)$,
which implies,

$$\min \quad \sqrt{a^\top \textstyle\sum_x a}+\sqrt{a^\top \textstyle\sum_y a} \quad \text{s.t. } a^\top(\overline{x}-\overline{y})=1 \tag{7}$$

# Minimax probability machine(contd.)

Any covariance matrix is a symmetric matrix and positive semi-definite.

If M is a positive semi-definite matrix, then there exists a matrix B such that :

$$M = B^{\top} B \tag{8}$$

So,

$$\sum_x = C^{\top} C \tag{9}$$

And since covariance matrix is positive semi-definite and symmetric, its root will also be symmetric, so $C^{\top} = C$ and thus $\sum_x^{\frac{1}{2}} = C$.
So, (7) can be written as:

$$\min \quad \sqrt{a^{\top} C^{\top} C a} + \sqrt{a^{\top} D^{\top} D a} \qquad \text{s.t. } a^{\top}(\overline{x} - \overline{y}) = 1 \tag{10}$$

# Minimax probability machine(contd.)

$$\underset{n\times n}{\begin{bmatrix} c \end{bmatrix}} \times \underset{n\times 1}{\begin{bmatrix} a \end{bmatrix}} = \underset{n\times 1}{\begin{bmatrix} \, \end{bmatrix}} \qquad \underset{1\times n}{\begin{bmatrix} a & b & c & \ldots \end{bmatrix}} \times \underset{n\times 1}{\begin{bmatrix} a \\ b \\ c \\ \vdots \end{bmatrix}} = a^2+b^2+c^2+\ldots$$

$$\min \quad \sqrt{(Ca)^\top Ca} + \sqrt{(Da)^\top Da} \qquad \text{s.t. } a^\top(\overline{x}-\overline{y}) = 1 \tag{11}$$

(11) can be written as:

$$\min \quad \left\|(\textstyle\sum_x)^{\frac{1}{2}} a\right\| + \left\|(\textstyle\sum_y)^{\frac{1}{2}} a\right\| \qquad \text{s.t. } a^\top(\overline{x}-\overline{y}) = 1 \tag{12}$$

This optimization problem is a second order cone program problem, after obtaining the optimal solution $a_*, b_*$ for a new data point z, if $a_*z \geq b_*$, z is classified as class x, otherwise z belongs to the class y.

# Deep Minimax probability machine

1. Now we combine the MPM with deep neural networks for the sake of their complementary strengths in the classification task and robustness learning.
2. MPM can directly minimize the maximum probability of misclassification with mean vectors and covariance matrices of the data considering the global structural information.

# Deep MPM (contd.)

Let $g(x, w)$ denotes a nonlinear mapping given by a deep neural network, parametrized by weight $w$. Through a neural network, we obtain effective representation for two classes of data $g(x, w)$ and $g(y, w)$ respectively, making mean vectors and covariance matrices reliable.

$$x \sim (\overline{x}, \sum_{x}) \rightarrow g(x, w) \sim \left(\overline{g(x, w)}, \sum_{g(x,w)}\right) \tag{13}$$

$$y \sim (\overline{y}, \sum_{y}) \rightarrow g(y, w) \sim \left(\overline{g(y, w)}, \sum_{g(y,w)}\right) \tag{14}$$

where $\overline{g(x, w)}, \overline{g(y, w)}$ denote mean vectors of two classes of data respectively, and $\sum_{g(x,w)}, \sum_{g(y,w)}$ denote covariance matrices of two classes of data respectively. For simplicity, we omit the parameter w of g($\cdot$) in further equations.

# Deep MPM (contd.)

We desire a hyperplane $a^\top g(z) = b$ that separates the two classes of data points with maximal probability given the means and covariance matrices obtaining by a deep neural network. The formulation of our model is

$$\max_{\alpha,a,b} \alpha \text{ s.t. } \inf \Pr\left(a^\top g(x) \geq b\right) \geq \alpha, \qquad (15)$$
$$\inf \Pr\left(a^\top g(y) \leq b\right) \geq \alpha.$$

On simplifying in the similar way that is done in MPM, we get

$$\min \quad \sqrt{a^\top \textstyle\sum_{g(x)} a} + \sqrt{a^\top \textstyle\sum_{g(y)} a} \quad \text{s.t. } a^\top(\overline{g(x)} - \overline{g(y)}) = 1 \qquad (16)$$

In order to train our model in an end-to-end fashion, we employ the Lagrangian multiplier method to perform optimization.

# Deep MPM (contd.)

With the introduction of a Lagrange multiplier $\lambda$, we can minimize the objective function,

$$\mathcal{L} = \sqrt{a^{\top} \sum_{g(x)} a} + \sqrt{a^{\top} \sum_{g(y)} a} - \lambda(a^{\top}(\overline{g(x)} - \overline{g(y)}) - 1) \tag{17}$$

Now, this is trained with back propagation in an end-to-end fashion, using the chain rule to calculate derivatives about all the parameters.The derivative of weight w is written as:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} =& \frac{\partial \mathcal{L}}{\partial \sum_{g(x)}} \frac{\partial \sum_{g(x)}}{\partial g(x)} \frac{\partial g(x)}{\partial w} + \frac{\partial \mathcal{L}}{\partial \sum_{g(y)}} \frac{\partial \sum_{g(y)}}{\partial g(y)} \frac{\partial g(y)}{\partial w} + \\ &+ \frac{\partial \mathcal{L}}{\partial \overline{g(x)}} \frac{\partial \overline{g(x)}}{\partial g(x)} \frac{\partial g(x)}{\partial w} + \frac{\partial \mathcal{L}}{\partial \overline{g(y)}} \frac{\partial \overline{g(y)}}{\partial g(y)} \frac{\partial g(y)}{\partial w} \end{aligned} \tag{18}$$

To optimise the model, Nesterov momentum version of mini-batch SGD is used.

# Deep MPM (contd.)

With the optimised parameters $w_*, a_*$, we obtain $b_*$ and $\alpha_*$ as,

$$b_* = a_*^\top \overline{g(x)} - \frac{\sqrt{a_*^\top \sum_{g(x)} a_*}}{\sqrt{a_*^\top \sum_{g(x)} a_*} + \sqrt{a_*^\top \sum_{g(y)} a_*}} \tag{19}$$

$$\alpha_* = \frac{1}{(\sqrt{a_*^\top \sum_{g(x)} a_*} + \sqrt{a_*^\top \sum_{g(y)} a_*})^2 + 1} \tag{20}$$

$\alpha_*$ represents worst case accuracy of the future data.In general, machine learning is fully data-driven, with the goal of maximizing the accuracy of the known data in the average sense, while our model is to maximize the accuracy of the future data in the worst sense, which is more robust.

# Experiments

We test our model on MNIST and CIFAR-10 datasets. We train ten binary classifiers with CNN and DeepMPM.

1. **MNIST**: Modified National Institute of Standards and Technology - is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.
2. **CIFAR-10**: Canadian Institute For Advanced Research - it contains 60,000 32×32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.
3. **CNNs**: CNNs are type of deep neural networks which work best on visual images, using an architecture of sliding filters and convolutional input layers.
4. **FGSM attack**: One of the first and most popular adversarial attacks to date is referred to as the Fast Gradient Sign Attack (FGSM).

# Experiment MNIST

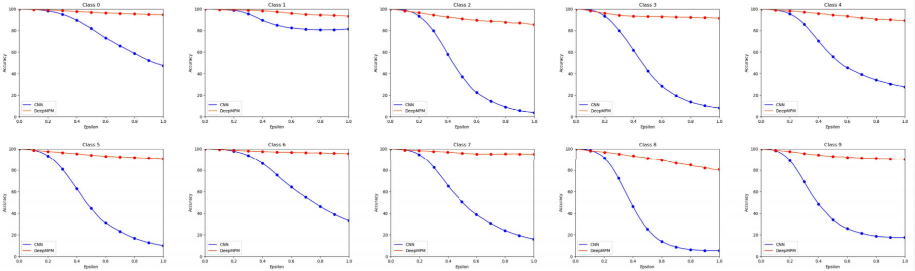

Figure: The accuracy of ten binary classifiers on MNIST for FGSM attacks. The horizontal axis represents the size of $\epsilon$. It's seen that the Accuracy of the DeepMPM decreases much more slowly with the size of adversarial perturbation for all ten classifiers. Thus, DeepMPM is more robust

# Experiment CIFAR-10

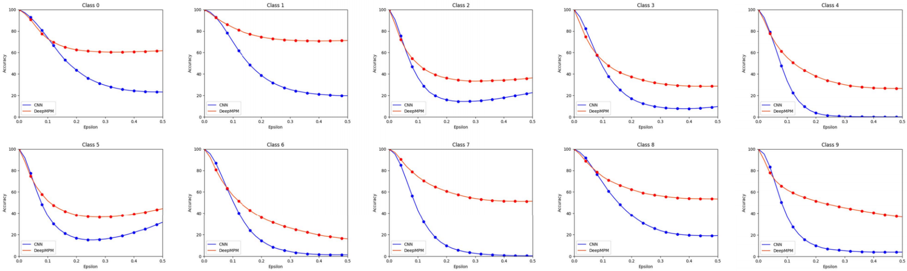

Figure: The accuracy of ten binary classifiers from the FGSM attacks on CIFAR-10. DeepMPM's accuracy decreases significantly slower with size of adversarial perturbation