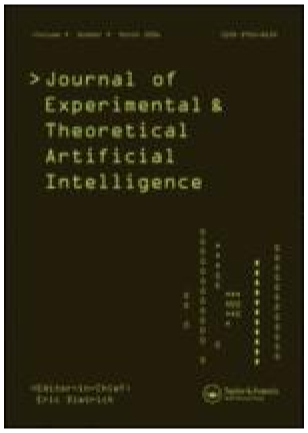

Explicit Matrix Gradient Expression for Residual Network

Journal:	*Journal of Experimental & Theoretical Artificial Intelligence*
Manuscript ID	TETA-2022-0228
Manuscript Type:	Original Article
Keywords:	ResNet, matrix gradient expression, backpropagation, saliency map, adversarial example

SCHOLARONE™
Manuscripts

Explicit Matrix Gradient Expression for Residual Network

For Peer Review Only

April 12, 2022

Abstract

Residual Network (ResNet) is a distinguished network structure in deep learning, and its layers can be profound. We theoretically explore the mathematics characteristic of the ResNet, in particular, to pay attention to the matrix gradient information, which is a powerful and straightforward mathematical tool for analyzing the properties of ResNet in ResNet, such as, the matrix gradients of the loss function w.r.t the input and the weight parameters, and the matrix gradient of the entry of the logits w.r.t the input. A theorem about the explicit matrix expression of gradients in Resnet is given in this work. We detail the theorem by the matrix derivative definition method and matrix differentiation method. We further provide explicit matrix expressions of some deep learning algorithms in ResNet, including backpropagation, gradient-based adversarial attacks, and gradient-based saliency maps. Furthermore, the reasons why the ResNet network works are analyzed.

Keyword: ResNet, matrix gradient expression, backpropagation, saliency map, adversarial example.

1 Introduction

With artificial intelligence rapid development, deep learning algorithms have been widely applied in computer vision [1, 2, 3], natural language processing [4], speech recognition [5, 6], medical science [7, 8], reinforcement learning [9, 10], biometrics [11], and diagnostics [12]. Driven by the increasing availability of computer hardware and deep learning frameworks, neural networks have improved performance. In paticular, they are applied in security-critical applications for instance surveillance systems [13], autonomous vehicles [14].

Neural networks have become state-of-the-art models for various machine learning tasks, and deeper networks often achieve solid empirical performance. A landmark example is a residual network (ResNet) [15], efficiently optimized even at considerable depths such as 1000 layers. However, there exists a gap between this empirical success and the theoretical understanding: ResNets can be trained to almost zero loss with standard stochastic gradient descent [16], yet it is known that more considerable depth leads to an increasingly non-convex landscape even with the presence of residual connections [17]. One of the things that made ResNet extraordinarily popular was the simple design strategy, which introduced only one identity shortcut. Despite the great success of identity shortcuts, subsequent work analyzed the weaknesses of identity shortcuts. The identity shortcut skips remaining blocks to preserve features and therefore may limit the representation capability of the network [18]. The disadvantage of identity shortcuts is that they can lead to domain collapse problems, thus reducing the learning ability of the network [19], and [20] suggested the use of nonlinear shortcuts to alleviate this problem.

In this work, the basic mathematics question of the explicit matrix expression of a gradient in ResNet is discussed. The explicit matrix form of the gradient is a powerful and straightforward mathematical tool for analyzing the properties of ResNet. Based on this view, we attempt to answer the following four questions:

- What is the explicit matrix form of the analytic expression for backpropagation (BP) in a ResNet?
- What is the explicit matrix form of the analytic expression for a gradient-based adversarial attack in a ResNet?
- What is the explicit matrix form of the analytic expression for the saliency map in a ResNet?
- Why can ResNet avoid the gradient vanishing or exploding?

These four questions can be summarized as one question: What are the explicit matrix forms of the analytic expressions for various parameter variables in a ResNet? An Answer to this problem through a strict mathematical proof is provided. Moreover, the main contributions of this work are summarized as follows:

- A theorem about the explicit expressions for matrix gradients in ResNet is given, Using two methods: matrix derivative definition and matrix differentiation.
- We provide the explicit matrix form of the analytic expression for BP in a ResNet.
- We provide the explicit matrix form of the analytic expression for the gradient-based adversarial attack on a ResNet.
- The explicit matrix form of the analytic expression for the saliency map in a ResNet is discussed.
- We explain the reason why a ResNet can avoid gradient vanishing and exploding.

For Peer Review Only

The remainder of this paper is organized as follows. The mathematics symbol description of this paper is illustrated in Section 2. In Section 3, we introduce a core theorem in this paper. The matrix derivative definition method and matrix differential method to prove the theorem is provided in Sections 4 and 5. The explicit matrix expressions of the BP matrix derivative, adversarial examples, and saliency map in ResNet are described in Sections 6, 7, and 8. Finally, in Section 9, we explain the reason why ResNet works from the matrix view.

2 Symbol description

Given a input vector $\mathbf{X} \in \mathbb{R}^{p_0 \times 1}$ with a ground truth label $y \in \mathbb{R}^{1\times 1}$, a fully connected neural network $F(\cdot)$ with L Resnet blocks which are $B_1(\cdot), \cdots B_L(\cdot)$, an activation function $g(\cdot)$(e.g., Sigmoid or ReLU function), a loss function f_y, We have

$$Y = f_y(B_L(g(B_{L-1}(\cdots B_2(g(B_1(\mathbf{X}))\cdots)))), \tag{1}$$

$$\hat{\mathbf{B}} = B_L(g(B_{L-1}(\cdots B_2(g(B_1(\mathbf{X}))\cdots))), \tag{2}$$

where $\hat{\mathbf{B}}$ indicates a predict vector. For each Resnet block, we have

$$\begin{array}{ll}
\mathbf{z}^1 = B_1(\mathbf{X}) = \mathbf{W}^{12}g(\mathbf{W}^{11}\mathbf{X}) + \mathbf{W}^{13}\mathbf{X}, & \mathbf{a}^1 = g(\mathbf{z}^1),\\
\mathbf{z}^2 = B_2(\mathbf{a}^1) = \mathbf{W}^{22}g(\mathbf{W}^{21}\mathbf{a}^1) + \mathbf{W}^{23}\mathbf{a}^1, & \mathbf{a}^2 = g(\mathbf{z}^2),\\
\vdots & \vdots\\
\mathbf{z}^{L-1} = B_{L-1}(\mathbf{a}^{L-1}) = \mathbf{W}^{L-1,2}g(\mathbf{W}^{L-1,1}\mathbf{a}^{L-1}) + \mathbf{W}^{L-1,3}\mathbf{a}^{L-1}, & \mathbf{a}^{L-1} = g(\mathbf{z}^{L-1}),\\
\mathbf{z}^L = B_L(\mathbf{a}^L) = \mathbf{W}^{L2}g(\mathbf{W}^{L1}\mathbf{a}^L) + \mathbf{W}^{L3}\mathbf{a}^L, & Y = f_L(\boldsymbol{z}^L).
\end{array} \tag{3}$$

where for any index $l \in \{1, \cdots, L\}$, $\mathbf{a}^l$ is the input vector of l-th Resnet block $B_l(\cdot)$, $\mathbf{z}^l$ is the output vector of l-th Resnet block $B_l(\cdot)$. $\mathbf{W}^{l1} \in \mathbb{R}^{p_{l'} \times p_{l-1}}$, $\mathbf{W}^{l2} \in \mathbb{R}^{p_{l'} \times p_{l-1}}$, $\mathbf{W}^{l3} \in \mathbb{R}^{p_l \times p_{l-1}}$ are the weight matrix in Resnet block $B_l(\cdot)$.

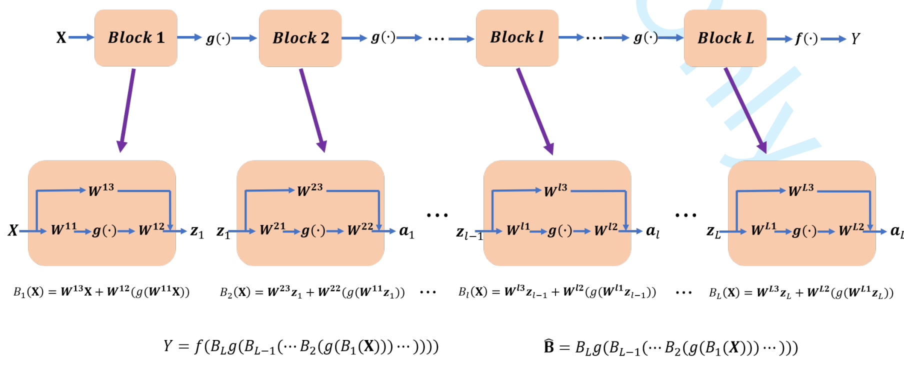
X
Block 1
g(·)
Block 2
g(·)
···
Block l
···
g(·)
Block L
f(·)
Y
W^{13}
X
W^{11}
g(·)
W^{12}
z_1
W^{23}
z_1
W^{21}
g(·)
W^{22}
a_1
···
W^{l3}
z_{l-1}
W^{l1}
g(·)
W^{l2}
a_l
···
W^{L3}
z_L
W^{L1}
g(·)
W^{L2}
a_L
B_1(X) = W^{13}X + W^{12}(g(W^{11}X))
B_2(X) = W^{23}z_1 + W^{22}(g(W^{11}z_1))
···
B_l(X) = W^{l3}z_{l-1} + W^{l2}(g(W^{l1}z_{l-1}))
···
B_L(X) = W^{L3}z_L + W^{L2}(g(W^{L1}z_L))
Y = f(B_L g(B_{L-1}(··· B_2(g(B_1(X)))···))))
B̂ = B_L g(B_{L-1}(··· B_2(g(B_1(X)))···)))

Figure 1: A schematic diagram of symbol description in a ResNet

For Peer Review Only

3 Theorem description

Theorem 1 *Given L Resnet blocks in a ResNet, Resnet block $B_l(\cdot)$ have weight matrices $\mathbf{W}^{l1} \in \mathbb{R}^{p_l \times p_{l'}}$, $\mathbf{W}^{l2} \in \mathbb{R}^{p_{l'} \times p_{l-1}}$, and $\mathbf{W}^{l3} \in \mathbb{R}^{p_l \times p_{l-1}}$, $l \in \{1, \cdots, L\}$. $\mathbf{X} \in \mathbb{R}^{p_0 \times 1}$ is a input vector, $Y \in \mathbb{R}^{1 \times 1}$ denotes a loss function. We have*

$$Y = f(B_L(g(B_{L-1}(\cdots B_2(g(B_1(\mathbf{X}))) \cdots)))),$$

$$\hat{\mathbf{B}} = B_L(g(B_{L-1}(\cdots B_2(g(B_1(\mathbf{X}))) \cdots))),$$

where $g(\cdot)$ represents an activation funtion, and $f(\cdot)$ indicates an loss function. Then the following five equations hold:

$$\frac{\partial Y}{\partial \mathbf{X}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^1 \mathbf{Q}^1)^\top \frac{\partial Y}{\partial \mathbf{a}^L}, \tag{4}$$

$$\frac{\partial \hat{\mathbf{B}}_r}{\partial \mathbf{X}} = (\mathbf{Q}^L_{[r-row]} \mathbf{D}^{L-1} \cdots \mathbf{D}^1 \mathbf{Q}^1)^\top, \tag{5}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l1}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^l \mathbf{W}^{l2} \hat{\mathbf{D}}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (\mathbf{a}^{l-1})^\top, \tag{6}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l2}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (g(\mathbf{W}^{l1} \mathbf{a}^{l-1}))^\top, \tag{7}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l3}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (\mathbf{a}^{l-1})^\top, \tag{8}$$

where r denotes the r-th entry of the vector $\hat{\mathbf{B}}_r$, and $[r - row]$ denotes the r-th row of a matrix. diag$\{\cdot\}$ *represents a diagonal matrix, and then we can define the matrices $\mathbf{Q}^i$, $\hat{\mathbf{D}}^i$, and $\mathbf{D}^i$ for i-th block in ResNet*

$$\begin{cases} \mathbf{Q}^i = \mathbf{W}^{i2} \hat{\mathbf{D}}^i \mathbf{W}^{i1} + \mathbf{W}^{i3}, \\ \hat{\mathbf{D}}^i = \mathrm{diag}\{g'(\mathbf{W}^{i1} \mathbf{a}^{i-1})\}, \\ \mathbf{D}^i = \mathrm{diag}\{g'(\mathbf{z}^i)\}, \\ i = 1, \cdots, L, . \end{cases}$$

The theorem above shows the explicit matrix expressions on the loss function w.r.t. the input vector and the weight parameters, and the entry of the logits vector w.r.t. the input vector in a ResNet, respectively. Moreover, using the theorem above, we derive the explicit matrix forms about crucial deep learning algorithms such as backpropagation, saliency map, and adversarial examples.

For Peer Review Only

4 Matrix Derivative Defination Method

4.1 The proof of Eq. (4)

According to Eq. (1) and Eq. (3), it is straightforward to know that the variables have the following dependence relationship between the scalar Y and the vector $\mathbf{X}$:

$$\mathbf{X} \to \mathbf{z}^1 \to \mathbf{a}^1 \to \mathbf{z}^2 \to \mathbf{a}^2 \to \cdots \to \mathbf{z}^L \to \mathbf{a}^L \to Y. \tag{9}$$

According to the matrix derivative definition method, the chain rule derivative of the scalar Y with respect to $\mathbf{X}_j$, can be computed as

$$\frac{\partial Y}{\partial \mathbf{X}_j} = \sum_{s_L=1}^{p_L} \frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} \frac{\partial \mathbf{a}^L_{s_L}}{\partial \mathbf{z}^L_{s_L}} \sum_{s_{L-1}=1}^{p_{L-1}} \frac{\partial \mathbf{z}^L_{s_L}}{\partial \mathbf{a}^{L-1}_{s_{L-1}}} \frac{\partial \mathbf{a}^{L-1}_{s_{L-1}}}{\partial \mathbf{z}^{L-1}_{s_{L-1}}} \cdots \sum_{s_1=1}^{p_1} \frac{\partial \mathbf{z}^2_{s_2}}{\partial \mathbf{a}^1_{s_1}} \frac{\partial \mathbf{a}^1_{s_1}}{\partial \mathbf{z}^1_{s_1}} \frac{\partial \mathbf{z}^1_{s_1}}{\partial \mathbf{X}_j}, \tag{10}$$

where $\mathbf{a}^i_{s_i}$ and $\mathbf{z}^i_{s_i}$ denote the s_i-th entry of the vector $\mathbf{a}^i$ and $\mathbf{z}^i$, $i \in 1, \cdots, L$, respectively.

$$\begin{cases} \dfrac{\partial \mathbf{z}^i_{s_i}}{\partial \mathbf{a}^{i-1}_{s_{i-1}}} = \sum_k \mathbf{W}^{i2}_{s_i k} g'(\sum_h \mathbf{W}^{i1}_{kh} \mathbf{a}^{i-1}) \mathbf{W}^{i1}_{k s_{i-1}} + \mathbf{W}^{i3}_{s_i s_{i-1}}, \\ \dfrac{\partial \mathbf{a}^i_{s_i}}{\partial \mathbf{z}^i_{s_i}} = g'(\mathbf{z}^i_{s_i}), \\ \mathbf{a}^0 = \mathbf{X}, \\ \hat{\mathbf{D}}^i = \mathrm{diag}\{g'(\mathbf{W}^{i1}\mathbf{a}^{i-1})\}, \\ \mathbf{D}^i = \mathrm{diag}\{g'(\mathbf{z}^i)\}, \\ i = 1, \cdots, L. \end{cases} \tag{11}$$

Plugging the results of Eq. (11) into Eq. (10), we can derive

$$\frac{\partial Y}{\partial \mathbf{X}_j} = \sum_{s_L=1}^{p_L} \frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} g'(\mathbf{z}^L_{s_L}) \sum_{s_{L-1}=1}^{p_{L-1}} \mathbf{Q}^2_{s_L s_{L-1}} g'(\mathbf{z}^{L-1}_{s_{L-1}}) \cdots \sum_{s_1=1}^{p_1=1} \mathbf{Q}^2_{s_2 s_1} g'(\mathbf{z}^1_{s_1}) \mathbf{Q}^1_{s_1 j}. \tag{12}$$

Assume that $\mathbf{M}^1_{js_2}$ denote the entry of the j-th row and s_2-th column of matrix $\mathbf{M}^1$, we have

$$\mathbf{M}^1_{js_2} = \sum_{s_1=1}^{p_1} \mathbf{Q}^2_{s_2 s_1} g'(\mathbf{z}^1_{s_1}) \mathbf{Q}^1_{s_1 j} = (\mathbf{Q}^1)^\top_{[j-row]} \mathbf{D}^1 (\mathbf{Q}^2)^\top_{[s_2-col]}, \tag{13}$$

where $[j-row]$ denotes j-th row of a matrix, and $[s_2-col]$ denotes s_2-th column of a matrix. From Eq. (13), it is straightforward to have

$$\mathbf{M}^1_{[j-row]} = (\mathbf{Q}^1)^\top_{[j-row]} \mathbf{D}^1 (\mathbf{Q}^2)^\top. \tag{14}$$

For Peer Review Only

Assume that $\mathbf{M}^2_{js_3}$ denote the entry of the j-th row and s_3-th column of matrix $\mathbf{M}^2$, we have

$$\mathbf{M}^2_{js_3} = \sum_{s_2=1}^{p_2} \mathbf{M}^1_{js_2} g'(\mathbf{z}^2_{s_2})\mathbf{Q}^2_{s_3 s_2} = \mathbf{M}^1_{[j-row]}\mathbf{D}^2(\mathbf{Q}^3)^\top_{[s_3-col]}. \tag{15}$$

From equation (15), we have

$$\mathbf{M}^2_{[j-row]} = (\mathbf{Q}^1)^\top_{[j-row]}\mathbf{D}^1(\mathbf{Q}^2)^\top\mathbf{D}^2(\mathbf{Q}^3)^\top. \tag{16}$$

$$\vdots$$

Assuming $\mathbf{M}^{L-1}_{js_L}$ denote the entry of the j-th row and s_L-th column of matrix $\mathbf{M}^{L-1}$, we have

$$\mathbf{M}^{L-1}_{js_L} = \sum_{s_{L-1}=1}^{p_{L-1}} \mathbf{M}^{L-2}_{js_{L-1}} g'(\mathbf{z}^{L-1}_{s_{L-1}})\mathbf{Q}^L_{s_L s_{L-1}} = \mathbf{M}^{L-2}_{[j-row]}\mathbf{D}^{L-1}(\mathbf{Q}^L)^\top_{[s_L-col]}. \tag{17}$$

From equation (17), it is straightforward to have

$$\mathbf{M}^{L-1}_{[j-row]} = (\mathbf{Q}^1)^\top_{[j-row]}\mathbf{D}^1(\mathbf{Q}^2)^\top\mathbf{D}^2(\mathbf{Q}^3)^\top\cdots(\mathbf{Q}^{L-1})^\top\mathbf{D}^{L-1}(\mathbf{Q}^L)^\top. \tag{18}$$

From equation (12) and (18), we have

$$\begin{aligned}\frac{\partial Y}{\partial \mathbf{X}_j} &= \sum_{s_L=1}^{p_L} \mathbf{M}^{L-1}_{js_L} g'(\mathbf{z}^L_{s_L})\frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} \\ &= (\mathbf{Q}^1)^\top_{[j-row]}\mathbf{D}^1(\mathbf{Q}^2)^\top\mathbf{D}^2(\mathbf{Q}^3)^\top\cdots(\mathbf{Q}^{L-1})^\top\mathbf{D}^{L-1}(\mathbf{Q}^L)^\top\frac{\partial Y}{\partial \mathbf{a}^L}.\end{aligned} \tag{19}$$

Further simplification of Eq. (19), so we can obtain

$$\frac{\partial Y}{\partial \mathbf{X}} = (\mathbf{D}^L\mathbf{Q}^L\mathbf{D}^{L-1}\cdots\mathbf{D}^1\mathbf{Q}^1)^\top\frac{\partial Y}{\partial \mathbf{a}^L}, \tag{20}$$

which completes the proof.

4.2 The proof of Eq. (5)

According to Eq. (2) and Eq. (3), it is straightforward to know that the variables have the following dependence relationship between the entry $\hat{\mathbf{B}}_\mathbf{r}$ of the vector $\hat{\mathbf{B}}$ and the vector $\mathbf{X}$:

$$\mathbf{X} \rightarrow \mathbf{z}^1 \rightarrow \mathbf{a}^1 \rightarrow \cdots \mathbf{z}^{L-1} \rightarrow \mathbf{a}^{L-1} \rightarrow \hat{\mathbf{B}}_\mathbf{r} \tag{21}$$

According to the matrix derivative definition, the chain rule derivation of the entry $\hat{\mathbf{B}}_\mathbf{r}$ with respect to $\mathbf{X}_j$, can be computed as

$$\frac{\partial \hat{\mathbf{B}}_r}{\partial \mathbf{X}_j} = \sum_{s_{L-1}=1}^{p_{L-1}} \frac{\partial \hat{\mathbf{B}}_r}{\mathbf{a}^{L-1}_{s_{L-1}}}\frac{\partial \mathbf{a}^{L-1}_{s_{L-1}}}{\partial \mathbf{z}^{L-1}_{s_{L-1}}} \sum_{s_{L-2}=1}^{p_{L-2}} \frac{\partial \mathbf{z}^{L-1}_{s_{L-1}}}{\partial \mathbf{a}^{L-2}_{s_{L-2}}}\frac{\partial \mathbf{a}^{L-2}_{s_{L-2}}}{\partial \mathbf{z}^{L-2}_{s_{L-2}}} \cdots \sum_{s_1=1}^{p_1} \frac{\partial \mathbf{z}^2_{s_2}}{\partial \mathbf{a}^1_{s_1}}\frac{\partial \mathbf{a}^1_{s_1}}{\partial \mathbf{z}^1_{s_1}}\frac{\partial \mathbf{z}^1_{s_1}}{\partial \mathbf{X}_j}. \tag{22}$$

For Peer Review Only

From equations (17) and (18), we have

$$\begin{aligned}\mathbf{M}^{L-2}_{js_{L-1}} &= \sum_{s_{L-2}=1}^{p_{L-2}} \frac{\mathbf{z}^{L-1}_{s_{L-1}}}{\mathbf{a}^{L-2}_{s_{L-2}}} \frac{\partial \mathbf{a}^{L-2}_{s_{L-2}}}{\partial \mathbf{z}^{L-2}_{s_{L-2}}} \cdots \sum_{s_1=1}^{p_1} \frac{\mathbf{z}^{2}_{s_2}}{\mathbf{a}^{1}_{s_1}} \frac{\partial \mathbf{a}^{1}_{s_1}}{\partial \mathbf{z}^{1}_{s_1}} \frac{\partial \mathbf{z}^{1}_{s_1}}{\partial \mathbf{X}_j} \\ &= (\mathbf{Q}^1)^\top_{[j-row]} \mathbf{D}^1 (\mathbf{Q}^2)^\top \mathbf{D}^2 (\mathbf{Q}^3)^\top \cdots (\mathbf{Q}^{L-1})^\top.\end{aligned} \tag{23}$$

Plugging the results of Eq. (23) into Eq. (22), we have

$$\begin{aligned}\frac{\partial \hat{\mathbf{B}}_r}{\partial \mathbf{X}_j} &= \sum_{s_{L-1}=1}^{p_{L-1}} \frac{\partial \hat{\mathbf{B}}_r}{\mathbf{a}^{L-1}_{s_{L-1}}} \frac{\partial \mathbf{a}^{L-1}_{s_{L-1}}}{\partial \mathbf{z}^{L-1}_{s_{L-1}}} \mathbf{M}^{L-2}_{js_{L-1}} = \sum_{s_{L-1}=1}^{p_{L-1}} \mathbf{Q}^L_{rs_{L-1}} g'(\mathbf{z}^{L-1}_{s_{L-1}}) \mathbf{M}^{L-2}_{js_{L-1}} \\ &= (\mathbf{Q}^1)^\top_{[j-row]} \mathbf{D}^1 (\mathbf{Q}^2)^\top \mathbf{D}^2 \cdots (\mathbf{Q}^{L-1})^\top \mathbf{D}^{L-1} (\mathbf{Q}^L)^\top_{[r-row]}.\end{aligned} \tag{24}$$

Finally, from Eq. (24), we can further get

$$\frac{\partial \hat{\mathbf{B}}_r}{\partial \mathbf{X}} = (\mathbf{Q}^L_{[r-row]} \mathbf{D}^{L-1} \cdots \mathbf{D}^1 \mathbf{Q}^1)^\top, \tag{25}$$

which completes the proof.

4.3 The proof of Eqs (6), (7) and (8)

According to equations (1) and (3), it is obvious to know the variables have the following dependence relationship between the scalar Y and the vector $\mathbf{W}^{lh}$ of the l-th Resnet Block:

$$\mathbf{W}^{lh} \to \mathbf{a}^l \to \mathbf{z}^{l+1} \to \mathbf{a}^{l+1} \to \cdots \to \mathbf{z}^L \to \mathbf{a}^L \to Y,$$

where $h \in \{1, 2, 3\}$, $l \in \{1, 2, \cdots, L\}$. According to Eq(1) and Eq. (3), it is straightforward to know the variables have the following dependence relationship between the scalar Y and the entry $\mathbf{W}^{lh}_{ij}$ of the vector $\mathbf{W}^{lh}$:

$$\frac{\partial Y}{\partial \mathbf{W}^{lh}_{ij}} = \sum_{s_L=1}^{p_L} \frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} \frac{\partial \mathbf{a}^L_{s_L}}{\partial \mathbf{z}^L_{s_L}} \sum_{s_{L-1}=1}^{p_{L-1}} \frac{\partial \mathbf{z}^L_{s_L}}{\partial \mathbf{a}^{L-1}_{s_{L-1}}} \frac{\partial \mathbf{a}^{L-1}_{s_{L-1}}}{\partial \mathbf{z}^{L-1}_{s_{L-1}}} \cdots \sum_{s_l=1}^{p_l} \frac{\partial \mathbf{z}^{l+1}_{s_{l+1}}}{\partial \mathbf{a}^l_{s_l}} \frac{\partial \mathbf{a}^l_{s_l}}{\partial \mathbf{z}^l_{s_l}} \frac{\partial \mathbf{z}^l_{s_l}}{\partial \mathbf{W}^{lh}_{ij}}. \tag{26}$$

Thus, from Eq. (3), it can easily obtain

$$\begin{cases} \dfrac{\partial \mathbf{z}^l_{sl}}{\partial \mathbf{W}^{l1}_{ij}} = \dfrac{\partial \sum\limits_k \mathbf{W}^{l2}_{s_l k} g(\sum\limits_n \mathbf{W}^{l1}_{kn} \mathbf{a}^{l-1}_n)}{\partial \mathbf{W}^{l1}_{ij}} = \mathbf{W}^{l2}_{s_l i} g'(\sum\limits_n \mathbf{W}^{l1}_{in} \mathbf{a}^{l-1}_n) \mathbf{a}^{l-1}_j, \\ \dfrac{\partial \mathbf{z}^l_{s_l}}{\partial \mathbf{W}^{l2}_{ij}} = g(\sum\limits_n \mathbf{W}^{l1}_{jn} \mathbf{a}^{l-1}_n) \delta_{s_l i}, \\ \dfrac{\partial \mathbf{z}^l_{s_l}}{\partial \mathbf{W}^{l3}_{ij}} = \mathbf{a}^{l-1}_j \delta_{s_l i}. \end{cases}, \tag{27}$$

For Peer Review Only

For Peer Review Only

where if $s_l = i$, then $\delta_{s_l i} = 1$; otherwise $\delta_{s_l i} = 0$. Plugging the results of Eq. (11) and Eq. (27) into Eq. (26), one can get equations

$$\frac{\partial Y}{\partial \mathbf{W}^{l3}_{ij}} = \sum_{s_L=1}^{p_L} \frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} g'(\mathbf{z}^L_{s_L}) \sum_{s_{L-1}=1}^{p_{L-1}} \mathbf{Q}^L_{s_L s_{L-1}} g'(\mathbf{z}^{L-1}_{s_{L-1}}) \cdots \sum_{s_l=1}^{p_l=1} \mathbf{Q}^{l+1}_{s_{l+1} s_l} g'(\mathbf{z}^l_{s_l}) \mathbf{a}^{l-1}_j \delta_{s_l i}, \tag{28}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l3}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (\mathbf{a}^{l-1})^\top, \tag{29}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l2}_{ij}} = \sum_{s_L=1}^{p_L} \frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} g'(\mathbf{z}^L_{s_L}) \sum_{s_{L-1}=1}^{p_{L-1}} \mathbf{Q}^L_{s_L s_{L-1}} g'(\mathbf{z}^{L-1}_{s_{L-1}}) \cdots \sum_{s_1=1}^{p_1=1} \mathbf{Q}^{l+1}_{s_{l+1} s_l} g'(\mathbf{z}^l_{s_l}) g(\sum_n \mathbf{W}^{l1}_{jn} \mathbf{a}^{l-1}_n) \delta_{s_l i}, \tag{30}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l2}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (g(\mathbf{W}^{l1} \mathbf{a}^{l-1}))^\top, \tag{31}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l1}_{ij}} = \sum_{s_L=1}^{p_L} \frac{\partial Y}{\partial \mathbf{a}^L_{s_L}} g'(\mathbf{z}^L_{s_L}) \sum_{s_{L-1}=1}^{p_{L-1}} \mathbf{Q}^L_{s_L s_{L-1}} g'(\mathbf{z}^{L-1}_{s_{L-1}}) \cdots \sum_{s_l=1}^{p_l=1} \mathbf{Q}^{l+1}_{s_{l+1} s_l} g'(\mathbf{z}^l_{s_l}) \mathbf{W}^{l2}_{s_l i} g(\sum_n \mathbf{W}^{l1}_{in} \mathbf{a}^{l-1}_n) \mathbf{a}^{l-1}_j, \tag{32}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l1}} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^l \mathbf{W}^{l2} \hat{\mathbf{D}}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (\mathbf{a}^{l-1})^\top, \tag{33}$$

which completes the proof.

5 Matrix Differential Method

5.1 The proof of Eq. (4)

According to the matrix differential definition, the differential dY of a scalar Y with respect to vector $\mathbf{X}$, can be computed as

$$dY = \left(\frac{\partial Y}{\partial \mathbf{X}}\right)^\top d(\mathbf{X}). \tag{34}$$

From equation (11), all $i \in \{1, \cdots, L\}$, we have

$$d(\mathbf{a}^i) = d(g(\mathbf{z}^i)) = g'(\mathbf{z}^i) \odot d(\mathbf{z}^i) = \mathbf{D}^i d(\mathbf{z}^i), \tag{35}$$

$$\begin{aligned} d(\mathbf{z}^i) &= d(\mathbf{W}^{i2} g(\mathbf{W}^{i1} \mathbf{a}^{i-1}) + \mathbf{W}^{i3} \mathbf{a}^{i-1}) \\ &= \mathbf{W}^{i2} g'(\mathbf{W}^{i1} \mathbf{a}^{i-1}) \odot d(\mathbf{W}^{i1} \mathbf{a}^{i-1}) + \mathbf{W}^{i3} d(\mathbf{a}^{i-1}) \\ &= \mathbf{W}^{i2} \hat{\mathbf{D}}^i \mathbf{W}^{i1} d(\mathbf{a}^{i-1}) + \mathbf{W}^{i3} d(\mathbf{a}^{i-1}) \\ &= (\mathbf{W}^{i2} \hat{\mathbf{D}}^i \mathbf{W}^{i1} + \mathbf{W}^{i3}) d(\mathbf{a}^{i-1}) \\ &= \mathbf{Q}^i d(\mathbf{a}^{i-1}), \end{aligned} \tag{36}$$

where typically $\odot$ denotes the Hadamard product. Plugging Eq. (35) and Eq. (36) into Eq. (34), it can easily obtain

$$dY = (\frac{\partial Y}{\partial \mathbf{a}})^\top (g'(\mathbf{z}^L) \odot (\mathbf{Q}^L(g'(\mathbf{z}^{L-1}) \odot (\cdots(\mathbf{Q}^2(g'(\mathbf{z}^2) \odot (\mathbf{Q}^1 d(\mathbf{X})))) \cdots)))). \tag{37}$$

Moreover, according to Eq. (37), one can get

$$dY = (\frac{\partial Y}{\partial \mathbf{a}})^\top (\mathbf{D}^L\mathbf{Q}^L\mathbf{D}^{L-1}\mathbf{Q}^{L-1}\cdots\mathbf{D}^2\mathbf{Q}^2\mathbf{D}^1\mathbf{Q}^1)d(\mathbf{X}), \tag{38}$$

$$\frac{\partial Y}{\partial \mathbf{X}} = (\mathbf{D}^L\mathbf{Q}^L\mathbf{D}^{L-1}\cdots\mathbf{D}^1\mathbf{Q}^1)^\top \frac{\partial Y}{\partial \mathbf{a}^L}, \tag{39}$$

which completes the proof.

5.2 The proof of Eq. (5)

According to the matrix differential definition, the differential $d\mathbf{B}_r$ of a scalar $\mathbf{B}_r$ with respect to vector $\mathbf{X}$ can be computed as

$$d\mathbf{B}_r = \left(\frac{\partial \mathbf{B}_r}{\partial \mathbf{X}}\right)^\top d(\mathbf{X}). \tag{40}$$

Plugging Eq. (35) and Eq. (36) into Eq. (40), we have

$$d\hat{\mathbf{B}}_r = \mathbf{Q}^L_{[r-row]}(g'(\mathbf{z}^{L-1})\odot(\mathbf{Q}^{L-1}(g'(\mathbf{z}^{L-2})\odot(\cdots(\mathbf{Q}^2(g'(\mathbf{z}^2)\odot(\mathbf{Q}^1 d(\mathbf{X})))) \cdots)))). \tag{41}$$

Similarly, according to Eq. (41), we have

$$d\hat{\mathbf{B}}_r = (\mathbf{Q}^L_{[r-row]}\mathbf{D}^{L-1}\mathbf{Q}^{L-1}\cdots\mathbf{D}^2\mathbf{Q}^2\mathbf{D}^1\mathbf{Q}^1)d(\mathbf{X}), \tag{42}$$

$$\frac{\partial \hat{\mathbf{B}}_r}{\partial \mathbf{X}} = (\mathbf{Q}^L_{[r-row]}\mathbf{D}^{L-1}\cdots\mathbf{D}^1\mathbf{Q}^1)^\top, \tag{43}$$

which completes the proof.

5.3 The proof of Eqs (6), (7), and (8)

According to the matrix differential definition, for $h \in \{1, 2, 3\}$ the differential dY of a scalar Y with respect to vector $\mathbf{W}^{lh}$, can be computed as for

$$dY = \left(\frac{\partial Y}{\partial \mathbf{W}^{lh}}\right)^\top d(\mathbf{W}^{lh}). \tag{44}$$

From equation (3),one can get

$$d(\mathbf{a}^l) = d(\mathbf{W}^{l2}g(\mathbf{W}^{l1}\mathbf{a}^{l-1}) + \mathbf{W}^{l3}\mathbf{a}^{l-1}). \tag{45}$$

Further, we can easily get

$$d(\mathbf{a}^l) = \mathbf{W}^{l2}\hat{\mathbf{D}}^l d(\mathbf{W}^{l1})\mathbf{a}^{l-1}, \tag{46}$$

For Peer Review Only

$$d(\mathbf{a}^l) = d(\mathbf{W}^{l2})g(\mathbf{W}^{l1}\mathbf{a}^{l-1}), \tag{47}$$

$$d(\mathbf{a}^l) = d(\mathbf{W}^{l3})\mathbf{a}^{l-1}. \tag{48}$$

Plugging Eq. (46) into Eq. (45), one can get

$$\mathrm{Tr}(dY) = \mathrm{Tr}[(\frac{\partial Y}{\partial \mathbf{a}^L})^\top \mathbf{D}^L\mathbf{Q}^L \cdots \mathbf{D}^l\mathbf{Q}^h\mathbf{W}^{l2}\hat{\mathbf{D}}^h d(\mathbf{W}^{l1})\mathbf{a}^{l-1}], \tag{49}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l1}} = (\mathbf{D}^L\mathbf{Q}^L \cdots \mathbf{D}^l\mathbf{Q}^l\mathbf{W}^{l2}\hat{\mathbf{D}}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L}(\mathbf{a}^{l-1})^\top. \tag{50}$$

Plugging Eq. (47) into Eq. (45), one can get

$$\mathrm{Tr}(dY) = \mathrm{Tr}[(\frac{\partial Y}{\partial \mathbf{a}^L})^\top \mathbf{D}^L\mathbf{Q}^L \cdots \mathbf{D}^l\mathbf{Q}^l d(\mathbf{W}^{l2})g(\mathbf{W}^{l1}\mathbf{a}^{l-1})], \tag{51}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l2}} = (\mathbf{D}^L\mathbf{Q}^L \cdots \mathbf{D}^l\mathbf{Q}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L}(g(\mathbf{W}^{l1}\mathbf{a}^{l-1}))^\top. \tag{52}$$

Plugging Eq. (48) into Eq. (45), It can easily be obtained that

$$\mathrm{Tr}(dY) = \mathrm{Tr}[(\frac{\partial Y}{\partial \mathbf{a}^L})^\top \mathbf{D}^L\mathbf{Q}^L \cdots \mathbf{D}^l\mathbf{Q}^l d(\mathbf{W}^{l3})\mathbf{a}^{l-1}], \tag{53}$$

$$\frac{\partial Y}{\partial \mathbf{W}^{l3}} = (\mathbf{D}^L\mathbf{Q}^L \cdots \mathbf{D}^l\mathbf{Q}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L}(\mathbf{a}^{l-1})^\top, \tag{54}$$

which completes the proof.

6 BP Matrix Derivation

The discovery of BP is considered as an essential milestone in the history of neural networks. The BP learning algorithm was first proposed by Werbos[21] in the 1970s. It was rediscovered by Rumelhart and McClelland [22] in 1986 and has been widely used ever since [23]. The theory of the BP algorithm is based on error-correcting learning rules that modify the network in a way that gradually reduces the network's error rate of the connection weights. This error is the difference between the actual network output and the desired output. Thus, the BP learning algorithm is included in the supervised learning paradigm because of the desired output pattern [24, 25]. The BP algorithm can be partitioned into the following three steps:

- The computation of a feed-forward stage;
- Backward propagation of the error to both the output layer and the hidden layers;
- Updating the weights of the network connections. The algorithm stops when the value of the error function is less than a predefined acceptable tolerance.

For Peer Review Only

Formally, we assume

$$\begin{cases} \mathbf{e}^L = \mathbf{D}^L \dfrac{\partial Y}{\partial \mathbf{a}^L}, \\ \mathbf{e}^{L-1} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1})^\top \dfrac{\partial Y}{\partial \mathbf{a}^L}, \\ \mathbf{e}^{L-2} = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \mathbf{Q}^{L-1} \mathbf{D}^{L-2})^\top \dfrac{\partial Y}{\partial \mathbf{a}^L}, \\ \qquad \vdots \\ \mathbf{e}^1 = (\mathbf{D}^L \mathbf{Q}^L \mathbf{D}^{L-1} \cdots \mathbf{D}^2 \mathbf{Q}^2 \mathbf{D}^1)^\top \dfrac{\partial Y}{\partial \mathbf{a}^L}. \end{cases} \tag{55}$$

According to Eq. (55) and Eq. (6) of Theorem 1, we have the following recurrence euqation to calculate gradient of Y with respect to $\mathbf{W}^{l1}$ is given as follows,

$$\frac{\partial Y}{\partial \mathbf{W}^{l1}} = (\mathbf{W}^{l2} \hat{\mathbf{D}}^l)^\top \mathbf{e}^l (\mathbf{a}^{l-1})^\top, \quad l = 1, \cdots, L. \tag{56}$$

According to Eq. (55) and Eq. (7) of Theorem 1, we have the following recurrence euqation to calculate gradient of Y with respect to $\mathbf{W}^{l2}$ is given as follows,

$$\frac{\partial Y}{\partial \mathbf{W}^{l2}} = \mathbf{e}^l (g(\mathbf{W}^{l1} \mathbf{a}^{l-1}))^\top, \quad l = 1, \cdots, L. \tag{57}$$

According to Eq. (55) and Eq. (8) of Theorem 1, we have the following recurrence euqation to calculate gradient of Y with respect to $\mathbf{W}^{l3}$ is given as follows,

$$\frac{\partial Y}{\partial \mathbf{W}^{l3}} = \mathbf{e}^l (\mathbf{a}^{l-1})^\top, \quad l = 1, \cdots, L. \tag{58}$$

For Peer Review Only

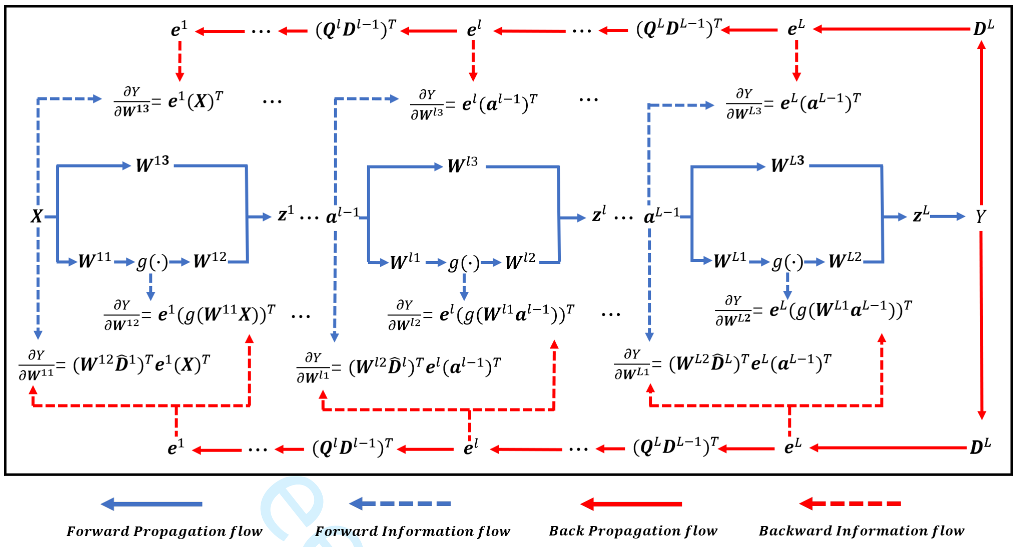

Figure 2: The overview of the matrix backpropagation process in the ResNet.

As shown in Figure 2, based on the iterative Eq. (56), Eq. (57) and (58), we give the overview of the matrix backpropagation process in the ResNet. The solid and dotted blue lines indicate the forward propagation and forward information flow in the ResNet, respectively. And the solid and dotted red ones indicate the backpropagation flow and the backward information flow in the ResNet, respectively. Figure 2 is straightforward to illustrate the calculation principle on the matrix gradient of weight parameters of each Resnet block.

7 Adversarial Examples

Adversarial attacks [26, 27, 28, 29] can be classified into the gradient, transferability/score, decision, and approximation according to the method used by the attack algorithm to generate the perturbation. In particular, gradient-based attacks approach are the most commonly used in the literature. The gradient-based algorithms use detailed information of the target model gradient w.r.t. the given input. This attack approach is usually performed in white-box scenarios when the attacker fully knows and has access to the targeted model.

The gradient-based adversarial attack includes FGSM, BIM, ILLCM, R+FGSM, and MI-FGSM. We can use equation (4) of Theorem 1 to give the matrix expression forms of the gradient-based adversarial attack in ResNet. According to the equation (4) of Theorem 1, we have

$$\nabla_{\mathbf{X}} f_y = \frac{\partial Y}{\partial \mathbf{X}} = (\mathbf{Q}^L\mathbf{D}^L\mathbf{Q}^{L-1}\cdots\mathbf{D}^2\mathbf{Q}^2\mathbf{D}^1\mathbf{Q}^1)^\top \frac{\partial Y}{\partial \mathbf{a}^L}, \tag{59}$$

$$\nabla_{\mathbf{X}} f_t = \frac{\partial Y}{\partial \mathbf{X}} = (\mathbf{Q}^L\mathbf{D}^L\mathbf{Q}^{L-1}\cdots\mathbf{D}^2\mathbf{Q}^2\mathbf{D}^1\mathbf{Q}^1)^\top \frac{\partial Y}{\partial \mathbf{a}^L}, \tag{60}$$

where $\nabla_{\mathbf{X}} f_y$ represents the gradient of the loss Y with respect to the input $\mathbf{X}$ with a ground truth label y, and $\nabla_{\mathbf{X}} f_t$ indicates the gradient of the loss Y with respect to the input $\mathbf{X}$ with a malicious targeted label t.

- **FGSM:** The fast gradient sign method (FGSM) proposed by Goodfellow et al. [26] was designed to quickly determine the direction for an adversarial perturbation for a given input sample. This perturbation increases the training loss of the target model and the possibility of inner class confusion, thereby reducing the classification confidence. According to the Eq. (59), we can get the explicit matrix gradient expression of the FGSM in ResNet as follows,
$$\mathbf{X}' = \mathbf{X} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{X}} f_y), \tag{61}$$
where $\mathbf{X}'$ represents an adversarial example. ε indicates the maximum adversarial perturbation, and $\mathrm{sign}(\cdot)$ indicates the signum function.

- **BIM**: The basic iterative method (BIM) proposed by Kurakin et al. [27] is one of the several enhancements of the FGSM and is sometimes called the iterative FGSM(I-FGSM). BIM is the first approach that was shown to be effective for printed paper examples. In addition to classification models, BIM has been used to attack semantic segmentation models. According to the Eq. (59), we can get the explicit matrix gradient expression of the BIM in ResNet as follows,
$$\begin{cases} \mathbf{X}'_0 = \mathbf{X}, \\ \mathbf{X}'_{i+1} = \mathrm{Clip}_{\varepsilon}\{\mathbf{X}'_i + \alpha \cdot \mathrm{sign}(\nabla_{\mathbf{X}} f_y)\}, \end{cases} \tag{62}$$
where $\mathbf{X}'_i$ is the adversarial variable, $\mathrm{Clip}_{\varepsilon}$ is the truncation function. α indicates the maximum adversarial perturbation at i step.

- **ILLCM**: The authors of BIM, Kurakin et al. [27], also proposed a targeted variant of the FGSM, named the iterative least likelihood class method (ILLCM), which aims to craft an adversarial example that is misclassified as a given target class. The ILLCM targets the class that is least likely to be selected by the original classifier. According to the Eq. (60), we can get the explicit matrix gradient expression of the ILLCM in ResNet as follows,
$$\begin{cases} \mathbf{X}'_0 = \mathbf{X}, \\ \mathbf{X}'_{i+1} = \mathrm{Clip}_{\varepsilon}\{\mathbf{X}'_i + \alpha \cdot \mathrm{sign}(\nabla_{\mathbf{X}} f_t)\}, \end{cases} \tag{63}$$

- **R+FGSM**: The random single-step attack (R+FGSM) proposed by Tramer et al. [30]. adds a small random perturbation to the input before applying an adversarial perturbation crafted by FGSM. This helps to circumvent the gradient masking defense strategy. According to the Eq. (59), we can get the explicit matrix gradient expression of the R+FGSM in ResNet as follows,
$$\begin{cases} \mathbf{X}^{\dagger} = \mathbf{X} + \alpha \cdot \mathrm{sign}((0,1)), \\ \mathbf{X}' = \mathbf{X}^{\dagger} + (\varepsilon - \alpha) \cdot \mathrm{sign}(\nabla_{\mathbf{X}} f_y), \end{cases} \tag{64}$$

For Peer Review Only

where $\mathbf{X}^{\dagger}$ indicates the random sample.

- **MI-FGSM**: Dong et al. [29] proposed a broad class of momentum-based iterative algorithms to boost adversarial attacks called the momentum iterative fast gradient sign method (MI-FGSM). Many gradient descent algorithms accelerate learning by accumulating a velocity vector in the gradient direction of the loss function across iterations. According to the Eq. (59), we can get the explicit matrix gradient expression of the MI-FGSM in ResNet as follows,

$$\begin{cases} \mathbf{X}'_0 = \mathbf{X}, \\ \boldsymbol{g}_{i+1} = \mu \cdot \boldsymbol{g}_i + \dfrac{\nabla_{\mathbf{X}} f_y}{\|\nabla_{\mathbf{X}} f_y\|_1}, \\ \mathbf{X}'_{i+1} = \mathbf{X}'_i + \alpha \cdot \operatorname{sign}(\boldsymbol{g}_{i+1}), \end{cases} \tag{65}$$

where $\|\cdot\|$ indicates the 1-norm function, and $\boldsymbol{g}_i$ indicates the i-th step momentum vector.

8 Saliency Map

A saliency map [31, 32, 33, 34, 35] is a visualization tool that can help humans understand and interpret the decisions made by deep neural networks (DNNs). In computer vision, generating intuitive heatmaps that highlight regions most related to a DNN's decision are an example of this technique. One common approach for determining salient regions relies on displaying changes in the model output with respect to its input images. Other approaches calculate gradients by back-propagating the prediction score through the target layer of the network and applying them as weights to generate the saliency maps.

Simonyan et al. [31] addressed the visualization of deep image classification models using a saliency map, and many researches subsequently studied on saliency maps. The method proposed by Simonyan et al. for computing the salient information of a known class in a given image (an image-specific class saliency map) uses a single backpropagation pass through the classification model. According to the euqation (5) of Theorem 1, we obtain the saliency map matrix expression of $\mathbf{X}$ with respect to the r-th element of the vector $\hat{\mathbf{B}}_r$,

$$\frac{\partial \hat{\mathbf{B}}_r}{\partial \mathbf{X}} = (\mathbf{Q}^L_{[r-row]}\mathbf{D}^{L-1}\cdots\mathbf{D}^1\mathbf{Q}^1)^\top. \tag{66}$$

Jacobian-based saliency map attacks (JSMA) proposed by Papernot et al. [28] extend saliency maps, previously introduced as visualization tools, to construct adversarial saliency maps. These maps indicate which input features an adversary should perturb in order to effect some desired change in a network output most efficiently. They are thus versatile tools that allow adversaries to generate large sets of adversarial examples. JSMA is a targeted attack based on the salience map. Maximal Jacobian-based saliency map attacks MJSMA,

For Peer Review Only

proposed by Wiyatno et al. [36], is an untargeted attack based on the saliency map. According to Eq. (66), we unify the matrix expression of JSMA and MJSMA:

$$\text{Map} = \frac{\partial \hat{\mathbf{B}}_o}{\partial \mathbf{X}} \odot \sum_{i \neq o} \frac{\partial \hat{\mathbf{B}}_i}{\partial \mathbf{X}}, \tag{67}$$

$$\text{Attack\_Map} = \begin{cases} \text{Map}[l], & \frac{\partial \hat{\mathbf{B}}_o}{\partial \mathbf{X}} \cdot \sum\limits_{i \neq o} \frac{\partial \hat{\mathbf{B}}_i}{\partial \mathbf{X}} < 0, \\ 0, & \text{otherwise} \end{cases}, \tag{68}$$

if o indicates a targeted class, the Attack_Map is a target map of JSMA, and then o indicates a source class, the Attack_Map is a untarget map of MJSMA.

9 Why does the ResNet works

When training is stable, assume that we have

$$B(\mathbf{a}^i) = \mathbf{a}^i, \quad i = L-k, \cdots, L. \tag{69}$$

We can have for back k blocks of the ResNet,

$$\begin{cases} \mathbf{W}^{L-k,1} = \mathbf{W}^{L-k+1,1} \cdots = \mathbf{W}^{L1} = \mathbf{0}, \\ \mathbf{W}^{L-k,2} = \mathbf{W}^{L-k+1,2} \cdots = \mathbf{W}^{L2} = \mathbf{0}, \\ \mathbf{W}^{L-k,3} = \mathbf{W}^{L-k+1,3} \cdots = \mathbf{W}^{L3} = \mathbf{E}, \end{cases} \tag{70}$$

where $\mathbf{E}$ denotes a identity matrix and $\mathbf{0}$ denotes a matrix which each entry equals 0. Thus, according to Eq. (6), Eq. (7) , and Eq. (8) of Theorem 1, we can calculate the limit of the matrix derivative norm about the scalar Y with respect to the vectors $\mathbf{W}^{l1}$, $\mathbf{W}^{l2}$, and $\mathbf{W}^{l3}$ in forward k blocks of the ResNet

$$\lim_{L\to\infty} \left\| \frac{\partial Y}{\partial \mathbf{W}^{l1}} \right\| = \left\| (\mathbf{D}^k \mathbf{Q}^k \mathbf{D}^{k-1} \cdots \mathbf{D}^l \mathbf{W}^{l2} \hat{\mathbf{D}}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (\mathbf{a}^{l-1})^\top \right\| < +\infty, \tag{71}$$

$$\lim_{L\to\infty} \left\| \frac{\partial Y}{\partial \mathbf{W}^{l2}} \right\| = \left\| (\mathbf{D}^k \mathbf{Q}^k \mathbf{D}^{k-1} \cdots \mathbf{D}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (g(\mathbf{W}^{l1} \mathbf{a}^{l-1}))^\top \right\| < +\infty, \tag{72}$$

$$\lim_{L\to\infty} \left\| \frac{\partial Y}{\partial \mathbf{W}^{l3}} \right\| = \left\| (\mathbf{D}^k \mathbf{Q}^k \mathbf{D}^{k-1} \cdots \mathbf{D}^l)^\top \frac{\partial Y}{\partial \mathbf{a}^L} (\mathbf{a}^{l-1})^\top \right\| < +\infty, \tag{73}$$

where L denotes the blocks number of the ResNet. We can find that $\lim\limits_{L\to\infty} \left\| \frac{\partial Y}{\partial \mathbf{W}^{l1}} \right\|$, $\lim\limits_{L\to\infty} \left\| \frac{\partial Y}{\partial \mathbf{W}^{l2}} \right\|$, and $\lim\limits_{L\to\infty} \left\| \frac{\partial Y}{\partial \mathbf{W}^{l3}} \right\|$ all are finite values. Therefore, with the increase of L, the ResNet will not have gradient vanish or explosion.

For Peer Review Only

10 Conclusion

In this paper, a theorem about the explicit matrix expression for gradients of ResNet is provided. The theorem is proven by using the matrix derivative definition and matrix differentiation. We further provide explicit matrix forms of some deep learning algorithms including backpropagation, gradient-based adversarial attacks, and gradient-based saliency maps. In addition, the reasons why the ResNet networks work are analyzed. In the future, we will study the explicit matrix expression for gradients about the different neural network structures, such as DesNet.

11 Acknowledge

This work is supported in part by the National Key R&D Program of China under Grant 2017YFB0802800.

References

References

[1] G. U. Xianfeng and X. Feng. Research on image classification algorithm based on deep unsupervised learning. *Journal of Pingdingshan University*, 2018.

[2] Dong Xu, Guo Lu, Ren Yang, and Radu Timofte. Learned image and video compression with deep neural networks. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–3. IEEE, 2020.

[3] Yannis Panagakis, Jean Kossaifi, Grigorios G Chrysos, James Oldfield, Mihalis A Nicolaou, Anima Anandkumar, and Stefanos Zafeiriou. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE*, 109(5):863–890, 2021.

[4] B Dalbelo Bašić and MP di Buono. An analysis of early use of deep learning terms in natural language processing. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 1125–1129. IEEE.

[5] Vukosi Rikhotso, Thipe Modipa, Madimetja Jonas Manamela, and Tumisho Bilson Mokgonyane. Development of automatic speech recognition for xitsonga using subspace gaussian mixture model. In *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–6. IEEE, 2021.

For Peer Review Only

[6] Yuxiang Kong, Jian Wu, Quandong Wang, Peng Gao, Weiji Zhuang, Yujun Wang, and Lei Xie. Multi-channel automatic speech recognition using deep complex unet. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 104–110. IEEE, 2021.

[7] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, pages 323–350, 2018.

[8] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[9] Rui Nian, Jinfeng Liu, and Biao Huang. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139:106886, 2020.

[10] Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109:101964, 2020.

[11] Ramar Ahila Priyadharshini, Selvaraj Arivazhagan, and Madakannu Arun. A deep learning approach for person identification using ear biometrics. *Applied Intelligence*, 51(4):2161–2172, 2021.

[12] Andreas Kleppe, Ole-Johan Skrede, Sepp De Raedt, Knut Liestøl, David J Kerr, and Håvard E Danielsen. Designing deep learning studies in cancer diagnostics. *Nature Reviews Cancer*, 21(3):199–211, 2021.

[13] Ali Tourani, Asadollah Shahbahrami, Sajjad Soroori, Saeed Khazaee, and Ching Yee Suen. A robust deep learning approach for automatic iranian vehicle license plate detection and recognition for surveillance systems. *IEEE Access*, 8:201317–201330, 2020.

[14] Sampurna Mandal, Swagatam Biswas, Valentina E Balas, Rabindra Nath Shaw, and Ankush Ghosh. Motion prediction for autonomous vehicles from lyft dataset using deep learning. In *2020 IEEE 5th international conference on computing communication and automation (ICCCA)*, pages 768–773. IEEE, 2020.

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

[16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

For Peer Review Only

[17] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Are deep resnets provably better than linear predictors? *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[19] G. Philipp, D Song, and J. G. Carbonell. Gradients explode - deep networks are shallow - resnet explained. In *International Conference on Learning Representations*, 2018.

[20] Chaoning Zhang, Francois Rameau, Seokju Lee, Junsik Kim, and In So Kweon. Revisiting residual networks with nonlinear shortcuts. In *British Machine Vision Conference (BMVC)*, 2019.

[21] P J Werbos. Beyond regression: New tools for prediction and analysis in the behavioral science. thesis (ph. d.). appl. math. harvard university. *Doctoral Dissertation Harvard University*.

[22] D.E. RUMELHART, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Readings in Cognitive Science*, 323(6088):399–421, 1988.

[23] Mutasem Khalil Sari Alsmadi, K. B. Omar, and S. A. Noah. Back propagation algorithm : The best algorithm among the multi-layer perceptron algorithm. *International journal of computer science and network security*, 9(4):378–383, 2009.

[24] Daniel Graupe. *Principles of artificial neural networks*, volume 7. World Scientific, 2013.

[25] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[26] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[27] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[28] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[29] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

For Peer Review Only

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

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[32] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[34] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[36] Rey Wiyatno and Anqi Xu. Maximal jacobian-based saliency map attack. *arXiv preprint arXiv:1808.07945*, 2018.

For Peer Review Only