UNDERSTANDING RESNET FROM A DISCRETE DYNAMICAL SYSTEM PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Residual network (ResNet) is one of popular networks proposed in recently years. Discussion about its theoretical properties is helpful for the understanding of networks with convolution modules. In this paper, we formulate the learning process of ResNet as a iterative system, then we may apply tools in discrete dynamical systems to explain its stability and accuracy. Due to the backward propagation of learning process, the module operations vary with the change of different layers. So we introduce the condition number of modules to describe the perturbation of output data, which can demonstrate the robustness of ResNet. In addition, the inter-class and intra-class median principal angle is defined to analyze the classification efficiency of ResNet. Mathematical description of the learning process of ResNet is given in a modular manner so that our research framework can be applied to other networks. In order to verify the feasibility of our idea, several experiments are carried out on the Dogs vs. Cats dataset, Kaggle: Animals-10 dataset, and ImageNet 2012 dataset. Simulation results are accordance with the theoretical analysis and prove the validity of our theory.

1 INTRODUCTION

Deep learning algorithms have become one of major trends in extracting image features for various vision tasks (Wu et al. (2021); Yun et al. (2020); Jiao & Zhao (2019)). Theoretical study of those algorithms can effectively explain the intrinsic principles of different applications (Wiatowski & Bölcskei (2017); Grohs et al. (2016); Chan et al. (2021); Higham & Higham (2019)), and provide guidance for the construction of suitable networks (Zhang et al. (2021); Liu et al. (2021); Li et al. (2021)).

Classification algorithms based on deep learning have great advantages of other traditional approaches in computer vision. People try to apply different theories (Bai et al. (2020); Su et al. (2018); Mengzi Tang (2021))to explore why those algorithms succeed. Qiu & Sapiro (2015) defined different classes as subspaces, and introduced a low-rank transformation of class subspaces to achieve target classification. Giryes et al. (2016) described the versatility of deep neural network (DNN) with random Gaussian weights in classification (Gulcu (2020)), which indicates that DNN is a general classifier based on the principal angle between two classes. Sokolic et al. (2017) proved that generalization error of neural network is decided by correlation among different weights. Bejani & Ghatee (2020) used the condition number of the weight matrix to evaluate the degree of overfitting. The larger the condition number, the more serious the overfitting. Xiao et al. (2019) utilized the condition number of the kernel matrix of an infinitely wide neural network as a measure of trainability. If the series of condition numbers diverges, the network will not be trained as the number of channels increases. Whether it is applied to evaluate the degree of overfitting or trainability, the larger condition number means that the learning network is ill-conditioned (Higham (2002)). In other words, this network is not suitable for the current task.

Above researches usually deal with general deep learning frameworks, which are not applicable directly for the real networks used in practice. To solve the problem, one starts theoretical analyses of specific networks including ResNet (He et al. (2016a); He et al. (2016b); Goceri (2019); Jagatap & Hegde (2019); Allen-Zhu & Li (2019)). Zhang & Schaeffer (2020) descripted the learning process of parameters as an optimization control problem. It reveals that the perturbation of the input data is influenced by the boundary of the learning parameter. Rousseau et al. (2020) classified the input data

and classification prediction of ResNet as two spaces respectively. Furthermore, a residual structure is described as a differential homeomorphism mapping from the input space to the prediction space. Ruthotto & Haber (2020) interpreted ResNet as a discrete space-time differential equation from the perspective of difference equations. He used this connection to analyze the stability of a new network similar to a difference system.

However, there are few discussions about the quantitative properties such as accuracy and stability of the deep learning networks. In fact, we may describe the learning process of ResNet as a discrete dynamical system. Since the iteration matrices constantly change in the learning process, we introduce the network condition number and the median principal angle to quantitatively investigate stability of ResNet parameter learning and accuracy of classification, respectively. The main contributions of this paper are summarized as follows: First, we apply ideas in discrete dynamical systems to numerically study the stability and accuracy of ResNet, and experiments show that the condition number and median principal angle concepts proposed in the paper efficiently explained the superiority of the ResNet algorithm. Second, the research framework based on the iteration equations can be actually generalized to any deep learning networks containing layers, which provides helpful guidance for the evaluation and construction of novel learning networks.

The remainder of our work is divided into the following parts. We introduce the basic network module and the ResNet module in section 2. Section 3 mainly gives the definition of condition number and makes the specific theoretical analysis that is based on ResNet. Section 4 denotes the median principal angle and shows experimental results of accuracy. We end with overall summary of the paper in section 5.

2 RELATED WORK

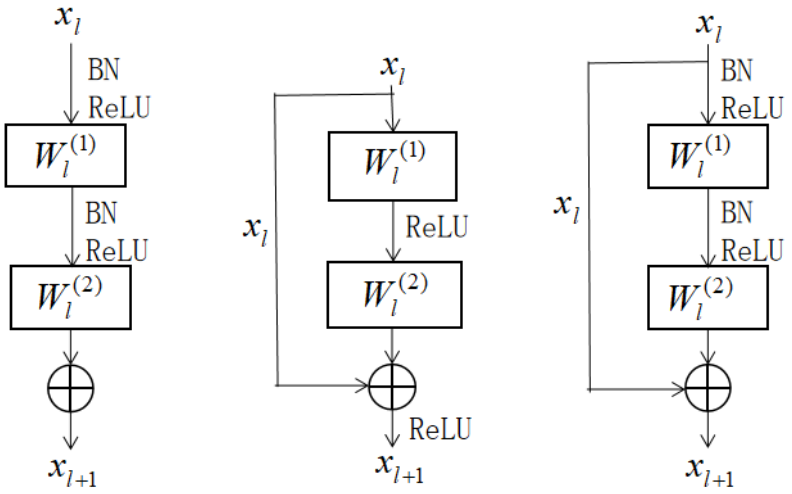
ResNet is the most popular deep convolution neural network (CNN) used in many application fields, such as (He et al. (2016a); Li & Rai (2020); Zhang (2021); Demir et al. (2019)). Its basic block consists of two convolutional layers and a shortcut connection (an identity connection). As the number of modules increases, the classification accuracy is significantly improved, and network robustness and generalization performance have also been improved. Subsequent improvements based on ResNet also result in efficient algorithms with high classification accuracy (Gao et al. (2019); Zagoruyko & Komodakis (2016); Radosavovic et al. (2020); Liang et al. (2018)). In order to theoretically explain advantages of ResNet, we successively introduce the basic network, original ResNet, and ResNet with BN.

Since the convolution operation in ResNet is linear, we may use matrix multiplication to represent it as follows. Assume that input vector $x \in \mathbb{R}^{n^2}$,the weight matrix $W \in \mathbb{R}^{n^2 \times n^2}$, and the characteristic output obtained after the convolution operation is $y \in \mathbb{R}^{n^2}$. Then the convolution process can be written as $y = W^T x$.

Basic block: As shown in Figure 1(a), the basic network we introduced here is the backbone of the convolutional neural network, that is, the input data is convolved to extract features. In order to compare it with ResNet, we constraint residual block with the same parameters in the basic network. In other words, the difference between basic block and ResNet block is only a shortcut connection. x_l is the input feature of the l-th module. $f : x_l \to x_{l+1}$ is the mapping of the input vector x_l to the output vector x_{l+1} in the basic network. the output of this layer is $x_{l+1} = f(x_l, \{W_l^{(i)}\}) = W_l^{(2)}\sigma(BN(W_l^{(1)}\sigma(BN(x_l))))$. where $W_l^{(i)}$ is the i-th weight in the l-th module, $i = 1, 2$. σ is the activation function (ReLU).

Basic ResNet block: The ResNet module is composed of the basic block and the identity mapping. Assume that input vector of ResNet in the l-th module is x_l, as shown in Figure 1(b), and the output vector of l-th module is x_{l+1} (i.e., the input of the next module). Then the output vector of the ResNet module is $x_{l+1} = \sigma[f(x_l, \{W_l^{(i)}\})] + x_l$, where $f(x_l, \{W_l^{(i)}\} = (W_l^{(2)})^T\sigma((W_l^{(1)})^T x_l)$.

Basic ResNet block added BN : As shown in Figure 1(c), the l-th module with BN in ResNet can be described as: The input feature of the l-th module is x_l, and the output of this layer is $x_{l+1} = f(x_l, \{W_l^{(i)}\}) + x_l$. Where $f(x_l, \{W_l^{(i)}\}) = (W_l^{(2)})^T\sigma(BN((W_l^{(1)})^T\sigma(BN(x_l))))$, BN is the batch normalization operation(Ioffe & Szegedy (2015)).



(a) a basic block (b) a basic ResNet block (c) a ResNet block with BN

Figure 1: Basic network mudule and ResNet modules. (a) is the backbone module of general convolutional neural networks, i.e., there are only convolution processing, batch normalization and nonlinear activation function. (b) denotes the ordinary ResNet module, which introduce ReLU activation function between two convolution kernels of the same size in a single module. (c) is that the input vector first perform batch normalization (BN), then pass activation fuction (ReLU) and convolution.

3 Stability analysis of ResNet

3.1 The condition number

Lemma 1 *Let $W_l^{(1)}, W_l^{(2)} \in \mathbb{R}^{n^2 \times n^2}$ be the two weight matrices of l-th module of ResNet. $I \in \mathbb{R}^{n^2 \times n^2}$ is the unit matrix, and $x_l \in \mathbb{R}^{n^2}$ is the input vector of l-th module. Then the corresponding weight matrix of whole module is $W_l^{(1)} W_l^{(2)} + I$, which satisfies the following inequality:*

$$(m_l^{(1)} m_l^{(2)} + 1)\|x_l\|_2 \leq \left\| (W_l^{(1)} W_l^{(2)} + I)^T x_l \right\|_2 \leq (M_l^{(1)} M_l^{(2)} + 1)\|x_l\|_2 \tag{1}$$

where $M_l^{(1)}, m_l^{(1)}$ are the maximum singular value and the minimum singular value of $W_l^{(1)}$, respectively, and $M_l^{(2)}, m_l^{(2)}$ are the maximum singular value and the minimum singular value of $W_l^{(1)}$, respectively.

The proof procedure of Lemma (1) is presented in Appendix A.

Remark1: According to inequality 1, we may define condition number of $W_l^{(1)} W_l^{(2)} + I$ is:

$$\kappa_{l,R}(W_l^{(1)} W_l^{(2)} + I) = \frac{M_l^{(1)} M_l^{(2)} + 1}{m_l^{(1)} m_l^{(2)} + 1}. \tag{2}$$

Remark2: Here weight matrix is always invertible. Assuming that the matrix is not full, the minimum singular value of the matrix is zero, and the condition number tends to infinity. Hence, the network diverges, which is contradictory to the actual classification network. Therefore, the matrix is full rank, and the singular values are non-negative.

3.2 Stability analysis of the ResNet block

Under the assumptions of equation(2), the condition number of l-th module of the base Network is $\kappa_{l,B}(W_l^{(1)} W_l^{(2)}) = \frac{M_l^{(1)} M_l^{(2)}}{m_l^{(1)} m_l^{(2)}}$. Apparently, we have $\kappa_{l,R}(W_l^{(1)} W_l^{(2)} + I) - \kappa_{l,B}(W_l^{(1)} W_l^{(2)}) < 0$.

That is to say, the ResNet network has a smaller condition numbers than the normal network in the condition of the same input features and the same convolutional kernel. Therefore, ResNet has faster convergent learning process in the forward propagation than Networks only based on convolution.

Next we try to discuss the evolutional property of ResNet. Take the $l\text{-}th$ module as example. Let $(W_l + I)^T(x_l + \Delta x_l) = a_l + \Delta a_l$, where Δx_l is the absolute error of the input vector of the $l\text{-}th$ module, a_l denotes the output vector of the $l\text{-}th$ module operation, Δa_l is the absolute error of the output vectoe after training. Then we present perturbation estimate of the output data of $l\text{-}th$ module in the following Lemma.

Lemma 2 *The perturbation of output data after the $l\text{-}th$ module operation satisfies the following inequality:*

$$\frac{1}{\kappa_{l,R}(W_l + I)} \cdot \frac{\|\Delta x_l\|}{\|x_l\|} \leq \frac{\|\Delta a_l\|}{\|a_l\|} \leq \kappa_{l,R}(W_l + I) \cdot \frac{\|\Delta x_l\|}{\|x_l\|}$$

The proof procedure of Lemma 2 is presented in Appendix B.

Similarly, following the same idea provided in Lemma 2, we can get the perturbation bounds about basic block. Let $W_l^T(x_l + \Delta x_l) = b_l + \Delta b_l$. x_l, Δx_l are the same as the case in ResNet. b_l, Δb_l are the output vector and perturbation of the output, respectively. Then, the relative error of the input data is limited to:

$$\frac{1}{\kappa_{l,B}(W_l)} \cdot \frac{\|\Delta x_l\|}{\|x_l\|} \leq \frac{\|\Delta b_l\|}{\|b_l\|} \leq \kappa_{l,B}(W_l) \cdot \frac{\|\Delta x_l\|}{\|x_l\|}.$$

Since $\kappa_{l,R}(W_l^{(1)}W_l^{(2)} + I) - \kappa_{l,B}(W_l^{(1)}W_l^{(2)}) < 0$, we can easily deduce that

$$\frac{1}{\kappa_{l,B}(W_l)} \cdot \frac{\|\Delta x_l\|}{\|x_l\|} \leq \frac{1}{\kappa_{l,B}(W_l+\mathrm{I})} \cdot \frac{\|\Delta x_l\|}{\|x_l\|}. \tag{3}$$

$$\kappa_{l,B}(W_l) \cdot \frac{\|\Delta x_l\|}{\|x_l\|} \geq \kappa_{l,R}(W_l + I) \cdot \frac{\|\Delta x_l\|}{\|x_l\|}. \tag{4}$$

From inequality (3), (4), we can find that the upper limit of relative error of the output data will be smaller and the lower limit will be larger in ResNet.

ResNet with BN has been shown to have advantage over base network in divergence. The following Lemma will prove that ResNet with BN indeed has smaller perturbation range than base network. i.e., the learning process of ResNet with BN is more stable with respect to the same input perturbation.

Let x_{l+1} is the actual output of $l\text{-}th$ module, y_{l+1} is the actual output vector of the $l\text{-}th$ module with BN, respectively. The loss function are $E = \frac{1}{2}(t_{l+1} - x_{l+1})^2$, $E_{BN} = \frac{1}{2}(t_{l+1} - y_{l+1})^2$, Respectively. σ denotes the ReLU activation function, and t_{l+1} denotes the expected output vector of $l\text{-}th$ module.

Lemma 3 *The output vectors of the ResNet module with BN and without BN can be written as $x_{l+1} = (W_l)^T\sigma(x_l)+x_l$, $y_{l+1} = (W_l)^T\sigma(BN(x_l))+x_l$, respectively. We have the following equality*

$$\frac{\|\Delta BN(x_l)\|}{\|BN(x_l)\|} \leq \frac{\|\Delta x_l\|}{\|x_l\|}.$$

The proof procedure of Lemma 3 is presented in Appendix C.

In fact, due to the iteration idea of the modules used in deep learning networks, the procedure analysis approach can be applicable for any other networks such as (Dosovitskiy et al. (2021); Cordonnier et al. (2020)).

4 ACCURACY ANALYSIS RESNET

4.1 THE MEDIAN PRINCIPAL ANGLE

Qiu & Sapiro (2015) applied the smallest principal angle to estimate inter-class classification result. Taking into account the influence of external noise, we introduce the median principal angle as the representation of classification. The larger the inter-class angle, the greater accuracy the classification. At the same time, we also introduce the intra-class median principal angle to illustrate the clustering effect. The smaller the intra-class median principal angle, the better effect the clustering.

Definition 1 *(Median principal angle) For two different matrices U, $V \in \mathbb{R}^{n^2 \times m}$, for any column vectors u_i, u_j of U, v_i, v_j of V, then the median principal angle between U and V is defined as*

$$\theta_{U,V} = \underset{u_i \in U, v_i \in V}{Med} \arccos \frac{u_i v_i}{\|u_i\|_2 \|v_i\|_2}.$$

The median principal angle of U is denoted by

$$\theta_U = \underset{u_i, u_j \in U}{Med} \arccos \frac{u_i u_j}{\|u_i\|_2 \|u_j\|_2}, i \neq j.$$

Among them, U and V are representing two different classes, the elements of those vectors represent different pixel values pictures. m is the number of examples of each class. Med (median) means to take the median of all principal angle.

For the median principal angle, we can also analyze the trend of the median principal angle during the learning process using the study in Section 3. However, in this paper we only demonstrate this rule by experiment, the theory interpretation will be represented in next work. We choose three different public datasets, Dogs vs. cats dataset, the kaggle: Animals 10 dataset, and the Imagenet 2012 dataset. For different requirements, we fix the epoch to be 50 and the batchsize to be 64 for each epoch when training the network, i.e., 3200 iterations for each network. We use intra- i to denote the median principal angle within the i class, and inter- i, j is denoted the median principal angle between the i and j classes.

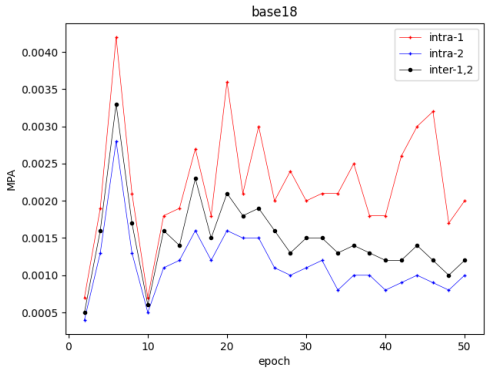
4.2 ACCURACY COMPARISON BETWEEN BASE NETWORK AND RESNET

Focusing on accuracy comparison of different Networks, we make three contrast experiments on Dogs vs. Cats data dataset. The experimental results are shown in Figure 2.

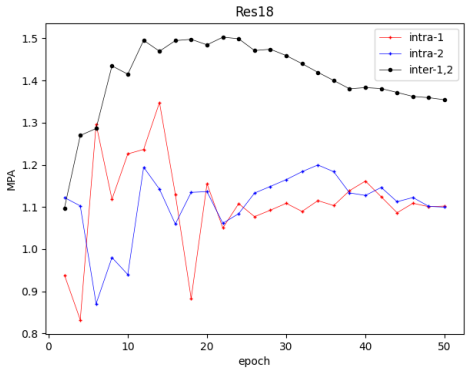
From the results in Figure 2, we can verify that our theoretical research is consistent with practical application. It can be seen from Figure 2(a) that the basic network 18 cannot accurately make clustering and classification. The classification accuracy rate is basically maintained at around 55% in experimental results. Comparing Figure 2(b) with Figure 2(c), it can be found that when BN is removed from the ResNet, the inter-class stability and intra-class stability decrease. Especially the perturbation in the interval of 5-20 epoch, the intra-class angle fluctuates greatly, which cause the training process instable. When the training is completed, the intra-class angle of the ResNet18 network without BN is significantly larger than the intra-class angle of the ResNet18 network with BN, i.e., the former clustering effect is not dominant. In addition, in the later stage of the iteration, the distance between the ResNet network classes without BN shows a decreasing trend, which shows that the classification accuracy rate will be relatively low. Therefore, the comparison between the two groups can show that the ResNet has relatively good results in accuracy and robustness.

Remark: In order to control the influence of different parameters on the network, the base network 18 in this article only represents the ordinary CNN network, which is different from the VGG (Simonyan & Zisserman (2014)) network structure, so it cannot be treated as VGG network.

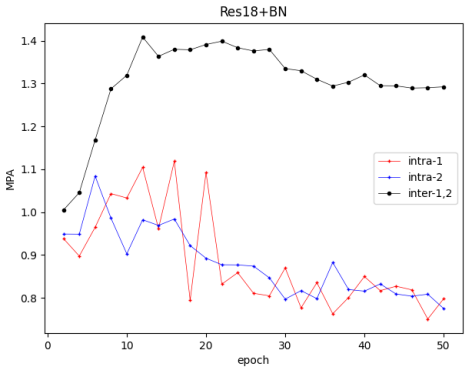
In order to compare the stability of different convolution layers of ResNet, we selected ResNet34, ResNet50, and ResNet101 for training in Dogs vs. Cats dataset. The experimental results are shown in Figure 3. Employing the same dataset to train different types of ResNet (different convolution layers), it is not necessarily that the more convolution layers, the better the stability and accuracy. For our small classification task, ResNet18 and ResNet34 have better classification stability and clustering effect.



(a) base18 iteration

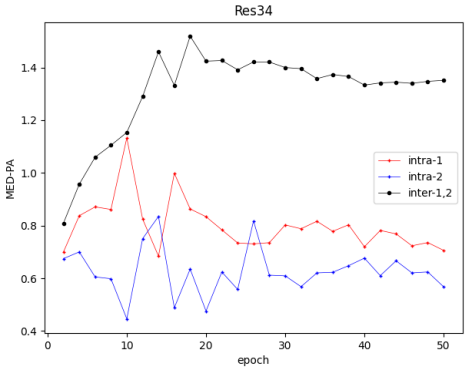


(b) Res18 iteration

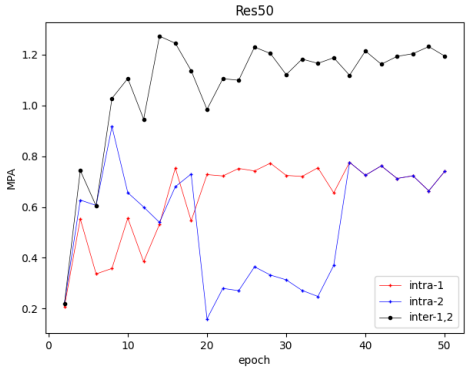


(c) Res18+BN iteration

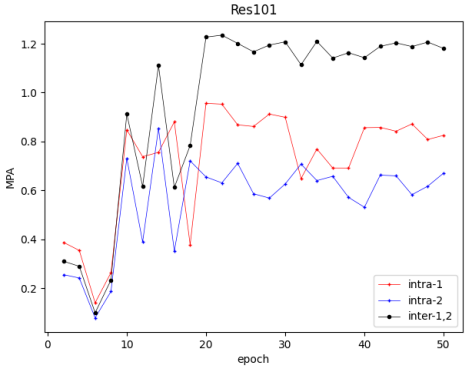
Figure 2: Comparison of the accuracy in different networks. (a) shows the comparison of the base network with 18 convolution layers, where the horizontal scale is the number of epochs, and the ordinate is the median principal angle (MPA). In-1 represents the median principal angle of the first class (Cats), in-2 represents the median principal angle of another class (Dogs), and out represents the median principal angle between cats and dogs; (b) represents the median principal angle when there is no BN in the ResNet18; (c) represents the median principal angle of ResNet18 that BN is added in this network.



(a) Res34 iterative process

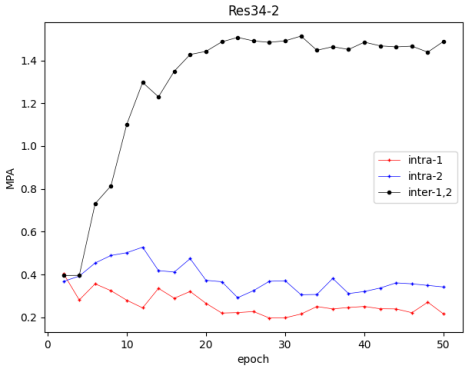


(b) Res50 iterative process

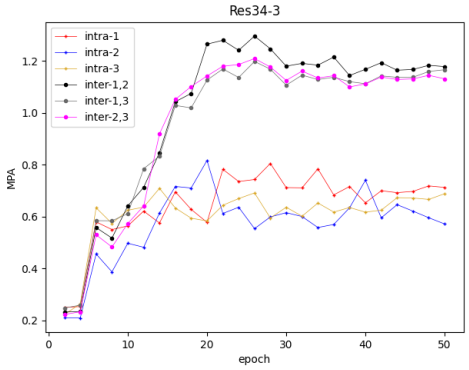


(c) Res101 iterative process

Figure 3: Comparison of the training iteration process of ResNet 34, ResNet 50, ResNet 101. (a) represents the training iteration process of ResNet34, intra-1, intra-2, and inter-1,2 are the same as those shown in Figure 2; (b) represents the training iteration process of ResNet50; (c) represents the training iteration process of ResNet101.



(a) 2-class in Animal 10 dataset

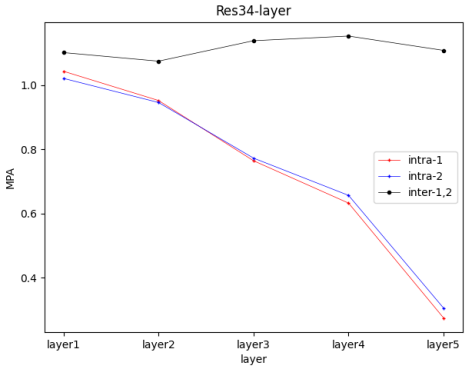


(b) 3-class in ImageNet-2012 dataset

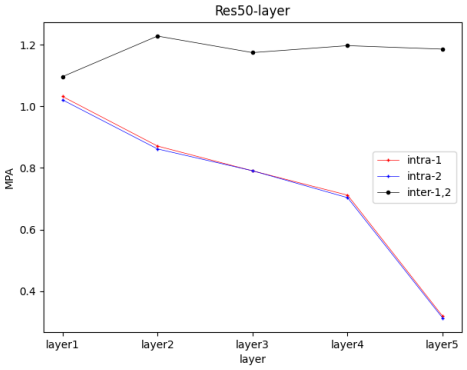
Figure 4: Comparison of the accuracy of different datasets on ResNet34. (a) represents the two-category iterative process of ResNet34 on the Animal 10 data set. (b) represents the three-category iterative process of ResNet34 on the Imagenet-2012 dataset, in-1, in-2, and in-3 represent the median principal in the first category, the second category, and the third category, out-1, out-2, out-3respectively represent the median principal angle between the first and second classes, the median principal angle between the first and third classes, and the second and third classes.

4.3 COMPARISON OF CLASSIFICATION ACCURACY OF DIFFERENT DATASETS

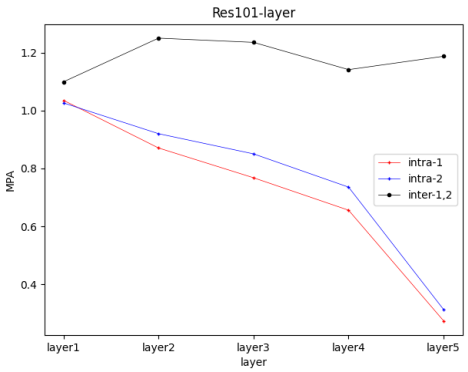
We illustrate the generalization of ResNet 34 in different datasets. We arbitrarily select two classes on the Animal 10 dataset for binary classification (except cats and dogs). At the same time, we do three-classification experiments in ImageNet-2012, where three different classes are randomly selected for training. The experimental results are shown in Figure 4.



(a) down-sampling in Res34



(b) down-sampling in Res50



(c) down-sampling in Res101

Figure 5: The trend of the median principal angle of the three different types of ResNet with different depths. (a) represents the changing trend of the median principal angle within and between classes in ResNet34 as the number of layers increases. (b) represents the changing trend of the median principal angle within and between classes in ResNet50 as the number of layers increases. (c) represents the changing trend of the median principal angle within and between classes in ResNet101 as the number of layers increases.

Combining Figure 3(a) and Figure 4, the classification accuracy of the same network is different in three different datasets. The training iterative process gradually stabilizes, and the same class and different classes can be clearly separated. Figure 3(a) and Figure 4(a) can show that, whether it is classification or clustering, the training iterative process is relatively stable, and the accuracy rate is also high. Figure 4(b) shows that the initial discontinuity of the network in the training is not obvious for the distinction within and between classes. However, the distinction within and between classes has become more and more obvious after epoch10, and the network has gradually stabilized afterwards.

Figure 4(a) and Figure 4(b) show that ResNet is suitable for different datasets, and the network has good stability and high accuracy. In addition, it is also suitable for multi-classification tasks, and the addition of classification tasks has little effect on its stability and accuracy.

4.4 COMPARISON OF CLASSIFICATION ACCURACY IN DIFFERENT LAYERS

4.1 and 4.2 describe the change process of the median principal angle between and within the class in iterations. In this section, we compare the change trend of the principal angle between and within the class for different layers. Different layers represent different numbers of channels. In ResNet34, ResNet50, and ResNet101, the number of layers is all 5 , which means that there are five different numbers of channels. In order to eliminate the factors that affect the accuracy of the number of iterations, we train these three networks on the Dogs vs. Cats dataset, with an iteration epoch of 50 and a batchsize of 64. In the testing phase, before each down-sampling, we extract the middle principal angle within and between classes. The experimental results are shown in Figure 5.

Figure 5 shows that as the number of layers increases, the intra-class median principal angle decreases more obviously, and the inter-class median principal angle increases slowly, but this does not affect the result of classification. After the last epoch, the distinction between the inter-class median principal angle and the intra-class median principal angle is especially obvious.

5 CONCLUSIONS

Inspired by the evolution idea of discrete dynamical systems, this paper has presented a set of evolution tools to investigate ResNet framework. Condition numbers of ResNet blocks are defined and analyzed to demonstrate the stability of ResNet. In addition, in order to further explain the accuracy of the network, we introduced median principal angles within and between classes to give an experimental proof of the superiority of ResNet.

In the next step, we plan to apply other theories of discrete dynamical systems, such as periodicity, bifurcation, etc. To analyze the evolution of the learning process based on different combinations of different network modules. In particular, we try to set up a systematic framework to evaluate the efficiency of various networks.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? *arXiv preprint arXiv:1905.10337*, 2019.

Chengzu Bai, Ren Zhang, Zeshui Xu, Baogang Jin, Jian Chen, Shuo Zhang, and Longxia Qian. Kernel low-rank entropic component analysis for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:5682–5693, 2020.

Mohammad Mahdi Bejani and Mehdi Ghatee. Theory of adaptive svd regularization for deep neural networks. *Neural Networks*, 128:33–46, 2020.

Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. Redunet: A white-box deep network from the principle of maximizing rate reduction. *arXiv preprint arXiv:2105.10446*, 2021.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. 2020.

Ahmet Demir, Feyza Yilmaz, and Onur Kose. Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3. pp. 1–4, 2019. doi: 10.1109/TIPTEKNO47231.2019.8972045.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Raja Giryes, Guillermo Sapiro, and Alex M. Bronstein. Deep neural networks with random gaussian weights: A universal classification strategy? *IEEE Transactions on Signal Processing*, 64(13): 3444–3457, 2016.

Evgin Goceri. Analysis of deep networks with residual blocks and different activation functions: classification of skin diseases. In *2019 Ninth international conference on image processing theory, tools and applications (IPTA)*, pp. 1–6. IEEE, 2019.

Philipp Grohs, Thomas Wiatowski, and Helmut B?lcskei. Deep convolutional neural networks on cartoon functions. pp. 1163–1167, 2016.

Talha Cihad Gulcu. Comments on deep neural networks with random gaussian weights: A universal classification strategy?. *IEEE Transactions on Signal Processing*, 68:2401–2403, 2020.

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*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Catherine F Higham and Desmond J Higham. Deep learning: An introduction for applied mathematicians. *Siam review*, 61(4):860–891, 2019.

Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pp. 448–456, 2015.

Gauri Jagatap and Chinmay Hegde. Linearly convergent algorithms for learning shallow residual networks. pp. 1797–1801, 2019. doi: 10.1109/ISIT.2019.8849246.

Licheng Jiao and Jin Zhao. A survey on the new generation of deep learning in image processing. *IEEE Access*, 7:172231–172263, 2019.

Xin Li and Laxmisha Rai. Apple leaf disease identification and classification using resnet models. pp. 738–742, 2020.

Zhiyuan Li, Yi Zhang, and Sanjeev Arora. Why are convolutional nets more sample-efficient than fully-connected nets? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=uCY5MuAxcxU`.

Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. pp. 2811–2820, 2018.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, and Zhaopeng Tu. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Bernard De Baets Mengzi Tang, Ral Prez-Fernndez. Distance metric learning for augmenting the method of nearest neighbors for ordinal classification with absolute and relative information. *Information Fusion*, 65:72–83, 2021. ISSN 1566-2535.

Qiang Qiu and Guillermo Sapiro. Learning transformations for clustering and classification. *J. Mach. Learn. Res.*, 16(1):187–225, 2015.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. pp. 10428–10436, 2020.

Franois Rousseau, Lucas Drumetz, and Ronan Fablet. Residual networks as flows of diffeomorphisms. *Journal of Mathematical Imaging and Vision*, 62(1), 2020.

Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62(3):352–364, 2020.

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

Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel R. D. Rodrigues. Generalization error of deep neural networks: Role of classification margin and data structure. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pp. 147–151, 2017.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. pp. 631–648, 2018.

Thomas Wiatowski and Helmut Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866, 2017.

Di Wu, Chao Wang, Yong Wu, Qi-Cong Wang, and De-Shuang Huang. Attention deep model with multi-scale deep supervision for person re-identification. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(1):70–78, 2021.

Lechao Xiao, Jeffrey Pennington, and Sam Schoenholz. Disentangling trainability and generalization in deep learning. 2019.

Sangseok Yun, Jae-Mo Kang, Il-Min Kim, and Jeongseok Ha. Deep artificial noise: Deep learning-based precoding optimization for artificial noise scheme. *IEEE Transactions on Vehicular Technology*, 69(3):3465–3469, 2020.

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

Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning robust state abstractions for hidden-parameter block mdps. 2021.

Linan Zhang and Hayden Schaeffer. Forward stability of resnet and its variants. *Journal of Mathematical Imaging and Vision*, 62(3):328–351, 2020.

Zhuoren Zhang. Resnet-based model for autonomous vehicles trajectory prediction. pp. 565–568, 2021.

A Appendix

The proof of lemma 1:

Let $(W_l^{(1)}W_l^{(2)} + I)$ be the weight matrix of the $l\text{-}th$ module of ResNet, its condition number is defined by $\kappa_{l,R}(W_l^{(1)}W_l^{(2)} + I) = \frac{M_l^{(1)}M_l^{(2)}+1}{m_l^{(1)}m_l^{(2)}+1}$. According to the definition in the numerical analysis, the condition number of this weight matrix is also denoted by

$$\kappa_{l,R}(W_l^{(1)}W_l^{(2)} + I) = \left\|W_l^{(1)}W_l^{(2)} + I\right\|_2 \cdot \left\|\left(W_l^{(1)}W_l^{(2)} + I\right)^{-1}\right\|_2.$$

In the $l\text{-}th$ module of ResNet, we approximately define

$$\left\|W_l^{(1)}W_l^{(2)} + I\right\|_2 = M_l^{(1)}M_l^{(2)} + 1, \left\|\left(W_l^{(1)}W_l^{(2)} + I\right)^{-1}\right\|_2 = \frac{1}{m_l^{(1)}m_l^{(2)}+1}.$$

Therefore, the following inequality is satisfied by

$$\begin{aligned}\left\|\left(W_l^{(1)}W_l^{(2)} + I\right)^T x_l\right\|_2 &\leq \left\|W_l^{(1)}W_l^{(2)} + I\right\|_2 \cdot \|x_l\|_2 \\ &= \left(M_l^{(1)}M_l^{(2)} + 1\right)\|x_l\|_2 .\end{aligned}$$

Another perspective to consider is that

$$\begin{aligned}\|x_l\|_2 &= \left\|(W_l^{(1)}W_l^{(2)}+I)^{-1}(W_l^{(1)}W_l^{(2)}+I)x_l\right\|_2 \\ &\leq \left\|(W_l^{(1)}W_l^{(2)}+I)^{-1}\right\|_2 \cdot \left\|(W_l^{(1)}W_l^{(2)}+I)x_l\right\|_2 .\end{aligned}$$

So the following inequality holds:

$$\begin{aligned}(m_l^{(1)}m_l^{(2)}+1)\|x_l\|_2 &= \frac{\|x_l\|_2}{\left\|(W_l^{(1)}W_l^{(2)}+I)^{-1}\right\|_2} \\ &\leq \left\|(W_l^{(1)}W_l^{(2)}+I)x_l\right\|_2 .\end{aligned}$$

Therefore, $(m_l^{(1)}m_l^{(2)}+1)\|x_l\|_2 \leq \left\|(W_l^{(1)}W_l^{(2)}+I)^T x_l\right\|_2 \leq (M_l^{(1)}M_l^{(2)}+1)\|x_l\|_2$.

lemma 1 is proved.

B APPENDIX

The proof procedure of lemma 2:

Let input vector be $x_l \in \mathbb{R}^{n^2}$, the process of training ResNet is

$$\begin{aligned}(W_l+I)^T(x_l+\Delta x_l) &= (W_l+I)^T x_l + (W_l+I)^T\Delta x_l \\ &= a_l + \Delta a_l .\end{aligned}$$

So following equations holds

$$(W_l+I)^T x_l = a_l$$

$$(W_l+I)^T\Delta x_l = \Delta a_l$$

$$\Delta x_l = ((W_t+I)^T)^{-1}\cdot(\Delta a_l)$$

Therefore,the norm of Δx_l satisfies:

$$\|\Delta x_l\| \leq \left\|(W_l+I)^{-1}\right\| \cdot \|\Delta a_l\| .$$

In other aspect,

$$\begin{aligned}\|a_l\| &= \left\|(W_l+I)^T \cdot x_l\right\| \\ &\leq \|W_l+I\| \cdot \|x_l\| ,\end{aligned}$$

so

$$\frac{\|\Delta x_l\|}{\|x_l\|} \leq \left\|(W_l+I)^{-1}\right\| \cdot \|W_l+I\| \cdot \frac{\|\Delta a_l\|}{\|a_l\|} = \kappa_{l,R}(W_l+I)\cdot\frac{\|\Delta a_l\|}{\|a_l\|}.$$

In other words,

$$\frac{\|\Delta a_l\|}{\|a_l\|} \geq \left\|(W_l+I)^{-1}\right\| \cdot \|W_l+I\| \cdot \frac{\|\Delta x_l\|}{\|x_l\|} = \frac{1}{\kappa_{l,R}(W_l+I)}\cdot\frac{\|\Delta x_l\|}{\|x_l\|}.$$

Also,

$$\|\Delta a_l\| = \left\|(W_l + I)^T \cdot (\Delta x_l)\right\| \leq \|W_l + I\| \cdot \|\Delta x_l\|,$$

the norm of Δx_l satisfies

$$\|\Delta x_l\| \geq \frac{\|\Delta a_l\|}{\|W_l+I\|},$$

so

$$\frac{\|\Delta x_l\|}{\|x_l\|} \geq \frac{\|\Delta a_l\|}{\|W_l+I\|\cdot\|x_l\|} \geq \frac{\|\Delta a_l\|}{\|W_l+I\|\cdot\left\|(W_l+I)^{-1}a_l\right\|} = \frac{1}{\kappa_{l,R}(W_l+I)} \cdot \frac{\|\Delta a_l\|}{\|a_l\|}.$$

In other words,

$$\frac{\|\Delta a_l\|}{\|a_l\|} \leq \kappa_{l,R}(W_l + I) \cdot \frac{\|\Delta x_l\|}{\|x_l\|}$$

In summary,

$$\frac{1}{\kappa_{l,R}(W_l+I)} \cdot \frac{\|\Delta x_l\|}{\|x_l\|} \leq \frac{\|\Delta a_l\|}{\|a_l\|} \leq \kappa_{l,R}(W_l + I) \cdot \frac{\|\Delta x_l\|}{\|x_l\|}.$$

lemma 2 is proved.

C APPENDIX

The proof process of Theorem 3 is as follows:

When no BN is added to l-th module, the module is represented as

$$x_{l+1} = (W_l)^T \sigma(x_l) + x_l,$$

the loss function is

$$E = \frac{1}{2}(t_{l+1} - x_{l+1})^2.$$

The gradient for W_l is

$$\frac{\partial E}{\partial W_l} = -(t_{l+1} - x_{l+1})\frac{\partial x_{l+1}}{\partial W_l} = (t_{l+1} - x_{l+1})(-\sigma(x_l)).$$

The new weight matrix is

$$(W_l - \eta \cdot \Delta W_l) = W_l - \eta \cdot \frac{\partial E}{\partial W_l}.$$

The learning process of network can be reinterpreted as

$$x_{l+1} + \Delta x_{l+1} = ((W_l)^T + (\Delta W_l)^T)x_l.$$

So

$$x_{l+1} = (W_l)^T x_l, \Delta x_{l+1} = (\Delta W_l)^T x_l.$$

The perturbation range of x_l is

$$\begin{aligned}
\frac{\|\Delta x_l\|}{\|x_l\|} &\leq \kappa_{l,R}(W_l + I) \cdot \frac{\|\Delta W_l\|}{\|W_l\|} \\
&= \kappa_{l,R}(W_l + I) \cdot \frac{\left\|-\eta\frac{\partial E}{\partial W_l}\right\|}{\|W_l\|} \\
&= \kappa_{l,R}(W_l + I) \cdot \frac{\|-\eta(t_{l+1} - x_{l+1})(-\sigma(x_l)\|}{\|W_l\|} \\
&= \kappa_{l,R}(W_l + I) \cdot \|\eta(t_{l+1} - x_{l+1})\| \cdot \frac{\|x_l\|}{\|W_l\|}.
\end{aligned}$$

When BN is added to l-th module, the module is represented as

$$y_{l+1} = (W_l)^T \sigma(BN(x_l)) + x_l,$$

its loss function is

$$E_{BN} = \frac{1}{2}(t_{l+1} - y_{l+1})^2.$$

One can calculate gradient for W_l, i.e.,

$$\frac{\partial E_{BN}}{\partial W_l} = -(t_{l+1} - y_{l+1})\frac{\partial y_{l+1}}{\partial W_l} = (t_{l+1} - y_{l+1})(-\sigma(BN(x_l))).$$

The new kernel matrix is

$$(W_l - \eta \cdot \Delta W_{l,BN}) = W_l - \eta \cdot \frac{\partial E_{BN}}{\partial W_l}.$$

The disturbance range of x_l is

$$\begin{aligned}
\frac{\|\Delta BN(x_l)\|}{\|BN(x_l)\|} &\leq \kappa_{l,R}(W_l + I) \cdot \frac{\|\Delta W_{l,BN}\|}{\|W_l\|} \\
&= \kappa_{l,R}(W_l + I) \cdot \frac{\left\|-\eta \frac{\partial E_{BN}}{\partial W_l}\right\|}{\|W_l\|} \\
&= \kappa_{l,R}(W_l + I) \cdot \frac{\|-\eta(t_{l+1} - y_{l+1})(-\sigma(BN(x_l))\|}{\|W_l\|} \\
&= \kappa_{l,R}(W_l + I) \cdot \|\eta(t_{l+1} - y_{l+1})\| \cdot \frac{\|BN(x_l)\|}{\|W_l\|}.
\end{aligned}$$

Suppose there exists M greater than zero, subject to

$$\max\{\|\eta(t_{l+1} - x_{l+1})\|, \|\eta(t_{l+1} - y_{l+1})\|\} \leq M.$$

Since $\|BN(x_l)\| \leq \|x_l\|$, we can directly deduce that

$$\kappa_{l,R}(W_l + I) \cdot M \cdot \frac{\|BN(x_l)\|}{\|W_l\|} \leq \kappa_{l,R}(W_l + I) \cdot M \cdot \frac{\|x_l\|}{\|W_l\|}.$$

Therefore, the relative error of input data with BN added is less than the relative error of input vector without BN, i.e.,

$$\frac{\|\Delta BN(x_l)\|}{\|BN(x_l)\|} \leq \frac{\|\Delta x_l\|}{\|x_l\|}.$$

Lemma3 is proofed.