An Interpretive Constrained Linear Model for ResNet and MgNet

Juncai He[a,*], Jinchao Xu[b,a], Lian Zhang[c], Jianqing Zhu[d]

[a]*Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, KAS*
[b]*Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA*
[c]*Shenzhen International Center for Industrial and Applied Mathematics, Shenzhen Research Institute of Big Data, Shenzhen 518172, China*
[d]*Faculty of Science, Beijing University of Technology, Beijing 100124, China*

Abstract

We propose a constrained linear data-feature-mapping model as an interpretable mathematical model for image classification using a convolutional neural network (CNN). From this viewpoint, we establish detailed connections between the traditional iterative schemes for linear systems and the architectures of the basic blocks of ResNet- and MgNet-type models. Using these connections, we present some modified ResNet models that, compared with the original models, have fewer parameters but can produce more accurate results, thereby demonstrating the validity of this constrained learning data-feature-mapping assumption. Based on this assumption, we further propose a general data-feature iterative scheme to demonstrate the rationality of MgNet. We also provide a systematic numerical study on MgNet to show its success and advantages in image classification problems, particularly in comparison with established networks.

Keywords: convolutional neural networks, data-feature mapping, multigrid iterative methods, ResNet, MgNet

*Corresponding author
 Email addresses: `juncai.he@kaust.edu.sa` (Juncai He), `xu@multigrid.org` (Jinchao Xu), `zhanglian@sribd.cn` (Lian Zhang), `jqzhu@emails.bjut.edu.cn` (Jianqing Zhu)
 URL: `https://juncaihe.github.io` (Juncai He), `http://www.personal.psu.edu/jxx1/` (Jinchao Xu)

1. Introduction

This paper focuses on providing mathematical insight into deep convolutional neural network (CNN) models that have been successfully applied in many machine learning and artificial intelligence areas such as computer vision, natural language processing, and reinforcement learning [1]. Examples of CNN models include the LeNet-5 model presented by LeCun et al. in 1998 [2], the AlexNet model by Hinton et al. in 2012 [3], the residual network (ResNet) by He et al. in 2015 [4], pre-act ResNet in 2016 [5], MgNet in 2019 [6], and other variants of CNN [7, 8, 9]. Among these CNN models, ResNet and pre-act ResNet are of special theoretical and practical interest. In fact, researchers have taken many steps to advance the field's theoretical understanding of ResNet, to explain how and why it works well, and to design better residual-type architecture based on empirical observations and informal interpretation [10, 11, 12, 13, 14, 15, 9]. Most of the works mentioned here focus on the interpretation of the basic block in ResNet. However, some fine structures of ResNet and MgNet remain unclear. For example, how do we interpret the convolutional kernels, what is the role of the activation function, and how do we explain the pooling operations? For the explanation of pooling operations in particular, as far as we are aware, almost no literature provides deep insights from a mathematical viewpoint. Given the natural multi-scale (multi-resolution) structure and the residual correction iterative scheme in multigrid methods [16, 17, 18], we were inspired and motivated to interpret MgNet and ResNet architecture (the whole feature extraction process including pooling layers) from the multigrid and iterative methods perspectives.

We propose a generic mathematical model underlying the basic blocks of ResNet and MgNet to demonstrate their dual relation and understand how they function. At the core of our model is the following assumption: there is a data-feature mapping

$$A * u = f, \tag{1.1}$$

where A is understood as the underlying data-feature mapping to be learned and in practice is implemented as a conversational kernel with multi-channel. In addition, f is the data such as images and u is the feature tensor such that

$$u \geq 0. \tag{1.2}$$

Feature extraction is then viewed as an iterative procedure (c.f. [16]) to solve (1.1):

$$u^i = u^{i-1} + B^i * (f - A * u^{i-1}), \quad i = 1 : \nu, \tag{1.3}$$

2

where ν is the total number of iterations. This is a typical residual correction iterative scheme for solving (1.1), where B^i is called the smoother which is also implemented as a conversational kernel with multi-channel in this work. Using, for example, the special activation function $\sigma(x) = \text{ReLU}(x) := \max\{0, x\} \geq 0$, the above iterative process can be naturally modified to preserve the constraint (1.2):

$$u^i = u^{i-1} + \sigma \circ B^i * \sigma(f - A * u^{i-1}), \quad i = 1 : \nu. \tag{1.4}$$

This forms the basic block of MgNet, precisely as in [6]. Given the dual relation of the solution and residual in solving linear systems from discrete partial differential equations (PDEs) [19, 18], we introduce this residual

$$r^i = f - A * u^i. \tag{1.5}$$

Now, the iterative process (1.4) can be written, in terms of the residual r^i as:

$$r^i = r^{i-1} - A * \sigma \circ B^i * \sigma(r^{i-1}), \quad i = 1 : \nu. \tag{1.6}$$

The iterative scheme (1.6) shares an almost identical structure with the basic block architecture of pre-act ResNet. Then, the analysis process shown above will be used to understand pre-act ResNet and to develop modified ResNet and pre-act ResNet models in this paper.

Furthermore, by drawing on the multigrid [16, 17] idea to restrict the residuals, we have a natural explanation for pooling operations in pre-act ResNet, which provides a basis for establishing a complete connection between pre-act ResNet and MgNet. Finally, we present numerical evidence to demonstrate that our constrained linear models (1.1) and (1.2) with the nonlinear iterative solver (1.4) or (1.6) provide a second interpretation and improvement on ResNet- and MgNet-type models. The main contributions of this paper can be summarized as follows:

- A constrained linear data-feature mapping is proposed and developed as an interpretable model to demonstrate the dual relation between ResNet- and MgNet-type models.

- Some natural modifications of ResNet-type models based on the constrained linear data-feature mapping are proposed.

- A general data-feature iterative scheme based on constrained linear data-feature mapping is proposed to show the rationality of MgNet.

- A systematic numerical study of MgNet is proposed to show its success in image classification problems and demonstrate its advantages over established networks in this context.

This paper is organized as follows. In Section 2, we review some related works. In Section 3, we introduce precise mathematical formulas to define ResNet-type and MgNet models. In Section 4, we propose the constrained linear data-feature mapping model to understand ResNet and MgNet architecture from the perspective of solving the constrained linear system based on theoretical observations and analysis. Then, we develop some modified ResNet models based on the constrained linear data-feature mapping presented. Finally, we propose a general data-feature iterative model to further demonstrate the rationality of MgNet. In Section 5, we demonstrate the validity of the constrained linear data-feature mapping assumption by comparing our modified ResNet-type models with the established ResNet-type models. In addition, we provide a systematic numerical study on MgNet. In Section 6, we offer some concluding remarks, including a brief discussion of the implications of the results reported herein and the investigative directions that can advance this research.

2. Related work

In [6], a unified neural network framework was proposed, known as MgNet, to establish the connections between ResNet-type CNNs and multigrid methods. In that work, the basic block of MgNet was first introduced, as in (1.4). These elementary components in block 1.4, including the residual term $f - A * u$, the convolutional operators A and B^i, the activation functions σ, and the positions of these two σ, were initially motivated by the deep connection between multigrid methods and ResNet. However, a natural interpretation of the underlying mechanism of the basic block iteration is still lacking (1.4). Furthermore, in this paper we propose the constrained linear model to interpret the basic block (1.4) from the iterative method perspective. Before MgNet, ideas and techniques from multigrid methods had been used to develop efficient CNNs. The researchers who developed ResNet [4] first took the multigrid methods as evidence to support what is known as a residual representation for the interpretation of ResNet. Further, [20, 21, 22] adopted multi-resolution ideas to improve the performance of their networks. Additionally, a CNN model with a structure similar to that of the V-cycle multigrid was proposed to address volumetric medical image segmentation and biomedical image segmentation in [23, 24]. The literature also includes studies focused on applying deep learning techniques in multigrid and numerical PDEs [25, 26].

Considering the connections between CNN models and some computational mathematics methods, researchers have also proposed the dynamic system or optimization perspective [21, 27, 28, 29, 30]. A key motivation of the dynamic systems viewpoint is that the iterative scheme $x^i = x^{i-1} + f(x^{i-1})$ in pre-act ResNet resembles the

4

forward Euler scheme in numerical dynamic systems. Following this idea, [31, 32] interpreted the data flow in ResNet as the solution of the transport equation in the characteristic line. Furthermore, [29] interpreted some different CNN models with residual block as some special discretization schemes for ordinary differential equations (ODEs), for example PloyNet [14], FractalNet [11], and RevNet [12]. Ignoring the specific discretization methods, [30] proposed a family of CNN models based on black-box solvers for ODEs. Some types of CNN architecture are further designed based on the iterative schemes of optimization algorithms [33, 34, 35]. These studies share the philosophy that many optimization algorithms can be considered as discretization schemes for some special ODEs [36].

Considering the resemble properties of ResNet, [37, 38] claim that ResNet is an ensemble of shallower models, and that discarding the intermediate residual block does not influence the model accuracy. [39, 40] point out that ResNet optimizes the risk in a functional space by combining an ensemble of effective features. In addition, some works, such as [41, 42, 43], propose to study the generalization and smoothness properties of ResNet from the Neural Tangent Kernel perspective. As for the approximation properties of ResNet, [44] demonstrates that a very deep ResNet with stacked modules, that have one neuron per hidden layer, and ReLU activation functions can uniformly approximate any Lebesgue integrable function. Recently, [45] studied and proved the approximation properties of ResNet and MgNet with multi-channel 3×3 kernels for functions with image-type inputs, i.e. functions defined on $\mathbb{R}^{d \times d}$.

3. Precise mathematical formulas for ResNet and MgNet

In this section, we introduce ResNet [4] and pre-act ResNet [5] with precise mathematical formulas. Then, we introduce MgNet [6] and its variants.

3.1. ResNet and Pre-act ResNet

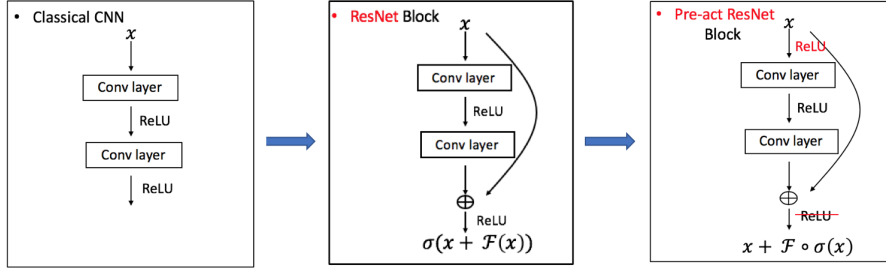Figure 3.1 demonstrates the connection and difference between classical CNN, ResNet [4], and pre-act ResNet [5].

5

Figure 3.1: Comparison of classical CNN, ResNet, and pre-act ResNet.

Here, $\sigma(x) = \mathrm{ReLU}(x) := \max\{0, x\}$ is the standard ReLU activation function. For ResNet and pre-act ResNet with the basic block $\mathcal{F}(x) = A * \sigma \circ B * x$, A and B are 3×3 convolutions with multichannel, zero padding, and stride one, and "$\circ$" means composition.

In order to investigate the interpretable mathematical model underlying these models, let us write these CNN models with precise mathematical formulas. The main structure of the pre-act ResNet without the last fully connected and soft-max layers can be written as in Algorithm 1.

Algorithm 1 $h = \text{pre-act ResNet}(f; J, \nu_1, \cdots, \nu_J)$

1: Initialization: $r^{1,0} = f_{\mathrm{in}}(f)$.
2: **for** $\ell = 1 : J$ **do**
3:     **for** $i = 1 : \nu_\ell$ **do**
4:         Basic Block:
$$r^{\ell,i} = r^{\ell,i-1} + A^{\ell,i} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}). \tag{3.1}$$
5:     **end for**
6:     Pooling(Restriction):
$$r^{\ell+1,0} = R_\ell^{\ell+1} *_2 r^{\ell,\nu_\ell} + A^{\ell+1,0} \circ \sigma \circ B^{\ell+1,0} *_2 \sigma(r^{\ell,\nu_\ell}). \tag{3.2}$$
7: **end for**
8: Final average pooling layer: $h = R_{\mathrm{ave}}(r^{J,\nu_J})$.

Here, $f_{\mathrm{in}}(\cdot)$ depends on the dataset and problems such as $f_{\mathrm{in}}(f) = \sigma \circ \theta^0 * f$ for CIFAR [46] and $f_{\mathrm{in}}(f) = R_{\max} \circ \sigma \circ \theta^0 *_2 f$ for ImageNet [47] as in [5]. Then, J denotes the total number of different spatial resolutions from $r^{1,0}$ to r^{L,ν_L}. More precisely, we have $J-1$ pooling operations from $r^{1,0}$ to r^{L,ν_L}. The indices ν_ℓ for $\ell = 1 : J$ denote the number of basic blocks (or iterations in the language of iterative methods or MgNet) with a fixed spatial resolution. In the basic block, $r^{\ell,i} = r^{\ell,i-1} + A^{\ell,i} * \sigma \circ B^{\ell,i} * \sigma(r^{i-1})$

6

is often called the basic ResNet block, where $A^{\ell,i}$ with $i \geq 0$ and $B^{\ell,i}$ with $i \geq 1$ are general 3×3 convolutions with zero padding and stride 1. In the pooling block (3.2), $*_2$ means the convolution with zero padding and stride 2; $R_\ell^{\ell+1}$ is taken as 1×1 kernel and referred to as the projection operator in MgNet [5]; and $B^{\ell,0}$ is taken as 3×3 convolutions, with the same channel dimension as the output channel dimension of $R_\ell^{\ell+1}$. During two consecutive pooling blocks, index ℓ refers to the fixed resolution or ℓ-th level grid as in the multigrid methods. In the final average pooling layer, R_{ave} means average pooling whereby the stride depends on the dataset and the problem considered.

The scheme of the original ResNet [4], which was actually developed earlier than pre-act ResNet, is very similar to that of pre-act ResNet but with a different order of convolutions and activation functions. For ResNet, the basic block and pooling operations are defined by

$$r^{\ell,i} = \sigma\left(r^{\ell,i-1} + A^{\ell,i} * \sigma \circ B^{\ell,i} * r^{\ell,i-1}\right), \tag{3.3}$$

$$r^{\ell+1,0} = \sigma\left(R_\ell^{\ell+1} *_2 r^{\ell,\nu_\ell} + A^{\ell+1,0} * \sigma \circ B^{\ell+1,0} *_2 r^{\ell,\nu_\ell}\right). \tag{3.4}$$

3.2. MgNet and its variants

In this subsection, we introduce the plain version of MgNet, and then discuss how to obtain variants of MgNet based on choosing different hyper-parameters in the plain MgNet.

3.2.1. Plain MgNet structure

Following the definitions and notations in [6], we show the plain version of MgNet in Algorithm 2.

Algorithm 2 $u^J = \text{MgNet}(f; J, \nu_1, \cdots, \nu_J)$

1: **Input**: number of grids J, number of smoothing iterations ν_ℓ for $\ell = 1 : J$, number of channels $c_{f,\ell}$ for f^ℓ and $c_{u,\ell}$ for $u^{\ell,i}$ on ℓ-th grid.

2: Initialization: $f^1 = f_{\text{in}}(f)$, $u^{1,0} = 0$

3: **for** $\ell = 1 : J$ **do**

4:     **for** $i = 1 : \nu_\ell$ **do**

5:         Feature extraction (smoothing):

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} * \sigma \left(f^\ell - A^\ell * u^{\ell,i-1} \right) \in \mathbb{R}^{c_{u,\ell} \times n_\ell \times m_\ell}. \tag{3.5}$$

6:     **end for**

7:     Note: $u^\ell = u^{\ell,\nu_\ell}$

8:     Interpolation and restriction:

$$u^{\ell+1,0} = \Pi_\ell^{\ell+1} *_2 u^\ell \in \mathbb{R}^{c_{u,\ell+1} \times n_{\ell+1} \times m_{\ell+1}}. \tag{3.6}$$

$$f^{\ell+1} = R_\ell^{\ell+1} *_2 \left(f^\ell - A^\ell * u^\ell \right) + A^{\ell+1} * u^{\ell+1,0} \in \mathbb{R}^{c_{f,\ell+1} \times n_{\ell+1} \times m_{\ell+1}}. \tag{3.7}$$

9: **end for**

Similar to ResNet, we consider $u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} * \sigma \left(f^\ell - A^\ell * u^{\ell,i-1} \right) \in \mathbb{R}^{c_{u,\ell} \times n_\ell \times m_\ell}$ to be the basic MgNet block. Here, $B^{\ell,i}$ with $i \geq 1$ are general 3×3 convolutions with zero padding and stride 1, which are interpreted as the smoother convolutions in multigrid. A^ℓ is also a 3×3 convolution with zero padding and stride 1 and is interpreted as the system operation as in the multigrid method. A key feature of MgNet that differs from the ResNet structure is that A^ℓ does not depend on the number of iterations on each grid. As discussed in [6], this can be understood as indicating that there is only one system to be solved on each grid. In interpolation and the restriction block (pooling block in ResNet), $*_2$ means convolution with zero padding and stride 2, $\Pi_\ell^{\ell+1}$ and $R_\ell^{\ell+1}$ are taken as 1×1 kernel.

3.2.2. Variants of MgNet based on different hyper-parameters

Based on the plain MgNet in Algorithm 2, it is natural to derive variants of MgNet by setting different hyper-parameter values. For simplicity, we use the following notation to represent different MgNet models with different hyper-parameters:

$$\text{MgNet}[\nu_1, \cdots, \nu_J]\text{-}[(c_{u,1}, c_{f,1}), \cdots, (c_{u,J}, c_{f,J})]\text{-}B^{\ell,i}. \tag{3.8}$$

These hyper-parameters are defined as follows:

- $[\nu_1, \cdots, \nu_J]$: The number of smoothing iterations on each grid. For example, $[2, 2, 2, 2]$ means that there are 4 grids and the number of iterations on each grid is 2.

8

- $[(c_{u,1}, c_{f,1}), \cdots, (c_{u,J}, c_{f,J})]$: The number of channels for $u^{\ell,i}$ and f^ℓ on each grid. We focus on the $c_{u,\ell} = c_{f,\ell}$ case, which suggests this simplification notation $[c_1, \cdots, c_J]$, or even $[c]$ if we further take $c_1 = c_2 = \cdots = c_J$. For example, MgNet[2, 2, 2, 2]-[64, 128, 256, 512] and MgNet[2, 2, 2, 2]-[256].

- $B^{\ell,i}$: This means that we use a different smoother $B^{\ell,i}$ in each smoothing iteration. Correspondingly, B^ℓ means that we share the smoother across all the grids:

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^\ell * \sigma \left(f^\ell - A^\ell * u^{\ell,i-1} \right). \tag{3.9}$$

Here, we note that we always use A^ℓ, which depends only on the grids.

For example, the notation MgNet[2, 2, 2, 2]-[256]-B^ℓ denotes an MgNet model with 4 different grids (feature resolutions), 2 smoothing iterations on each grid, 256 channels for both the feature tensor $u^{\ell,i}$ and the data tensor f^ℓ, and (3.9) as the smoothing iteration.

4. Constrained linear data-feature mapping

In this section, we establish a new understanding of pre-act ResNet and MgNet by drawing on the idea that the pre-act ResNet block and MgNet block are iterative schemes for solving some hidden model in each grid in a dual relation. Then, we adopt this assumption for the ResNet-type models and obtain some modified models with a special parameter-sharing scheme.

4.1. Constrained linear data-feature mapping and iterative methods

Here, we introduce the data and feature space of CNN, which is analogous to the function space and its duality in the theory of multigrid methods [18]. Specifically, following [6] we introduce the next data-feature mapping model in every grid level as follows:

$$A^\ell * u^\ell = f^\ell, \tag{4.1}$$

where f^ℓ and u^ℓ belong to the data and feature space of the ℓ-th grid. We now make the following two important observations for this data-feature mapping:

- The mapping in (4.1) is linear. More specifically, it is simply a convolution with multichannel, zero padding, and stride one as in pre-act ResNet or MgNet.

- In each level, namely between two consecutive poolings, there is only one data-feature mapping. Or, we say that A^ℓ depends only on ℓ, but not on the number of layers.

9

The assumption that this linear data-feature mapping depends only on the grid level ℓ is motivated by a basic property of multigrid methods [16, 17, 18].

In addition to (4.1), we introduce an important constrained condition in feature space whereby

$$u^{\ell,i} \geq 0. \tag{4.2}$$

The rationality of this constraint in feature space can be interpreted as follows. First, from the real neural system, the real neurons will only be active if the electric signal is greater than a certain threshold value, i.e. human brains can only see features with a certain threshold. On the other hand, the "shift" invariant property of feature space in CNN models, namely, $u + a$, will not change the classification results. This means that $u + a$ should have the same classification result as u. That is, we can assume $u \geq 0$ to reduce the redundancy of u.

Based on these assumptions, the next step is to solve the data-feature mapping equation in (4.1). We adopt some classical iterative methods [16] in scientific computing to solve the system (4.1) and obtain

$$u^{\ell,i} = u^{\ell,i-1} + B^{\ell,i} * (f^\ell - A^\ell * u^{\ell,i-1}), \ i = 1 : \nu_\ell, \tag{4.3}$$

where $u^\ell \approx u^{\ell,\nu_\ell}$. For a more detailed account of iterative methods in numerical analysis, we refer to [16, 48, 49]. To preserve (4.2), we naturally use the ReLU activation function σ to modify (4.3) as follows:

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} * \sigma(f^\ell - A^\ell * u^{\ell,i-1}), \quad i = 1 : \nu_\ell, \tag{4.4}$$

which is exactly the same as the basic block in MgNet as in Algorithm 2.

Because of the linearity of convolution in data-feature mapping, if we consider the residual $r^{\ell,j} = f^\ell - A^\ell * u^{\ell,j}$, (4.4) leads to the next iterative forms for the residuals:

$$r^{\ell,i} = r^{\ell,i-1} - A^\ell * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}). \tag{4.5}$$

This is the same as (3.3) under the constraint $A^{\ell,i} = A^\ell$ in pre-act ResNet.

We summarize the above derivation in the following theorem.

Theorem 4.1. *Under the assumption that there is only one linear data-feature mapping in each grid ℓ, i.e., $A^{\ell,i} = A^\ell$, the iterative form in feature space as in (4.3) is equivalent to (4.5) if A^ℓ is invertible where $r^{\ell,i} = f^\ell - A^\ell * u^{\ell,i}$.*

Here, we point out that the assumption, there is only one linear data-feature mapping on each level (sharing A), is critical to deriving the residual iterative form (3.3). Theoretically, the assumption can be explained from the approximation perspective.

A recent work [45] studied the approximation properties for plain CNNs, (pre-act) ResNet, and also MgNet. In the proofs of approximation properties for (pre-act) ResNet and MgNet in [45], the flexibility of the convolution kernel $A^{\ell,i}$ is much lower than $B^{\ell,i}$. This indicates that it should be reasonable to include extra assumptions on $A^{\ell,i}$ in the original ResNet and pre-act ResNet architecture.

4.2. Modified pre-act ResNet and ResNet

In this subsection, we propose some modified ResNet and pre-act ResNet models based on the assumption of the constrained linear data-feature mapping underlying these models. Although the scheme in (4.5) is closely related to the original pre-act ResNet, there is a major difference between the two given that in (4.5) there is an extra constraint, i.e., $A^{\ell,i} = A^\ell$. As a result, we obtain the next modified pre-act ResNet as follows:

Modified Pre-act ResNet (Pre-act ResNet-A^ℓ-$B^{\ell,i}$).

$$r^{\ell,i} = r^{\ell,i-1} + A^\ell * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}). \tag{4.6}$$

Here, we make a small modification to the sign before A^ℓ in the formula because of the linearity of convolution. As discussed, the modified pre-act ResNet model is derived from constrained linear data-feature mapping by using a special iterative scheme. Although we cannot obtain these connections in ResNet directly, formally we can make the modification from $A^{\ell,i}$ to A^ℓ into (3.1) to obtain the corresponding modified ResNet models:

Modified ResNet (ResNet-A^ℓ-$B^{\ell,i}$).

$$r^{\ell,i} = \sigma \left(r^{\ell,i-1} + A^\ell * \sigma \circ B^{\ell,i} * r^{\ell,i-1} \right). \tag{4.7}$$

A unified but simple diagram for how we change a pre-act ResNet to a modified pre-act ResNet is shown in Figure 4.1. The modified ResNet structure can be obtained correspondingly.

According to Theorem 4.1, the constrained linear model provides a precise interpretation and understanding of the dual relation between the MgNet model and the modified pre-act ResNet model. Briefly, MgNet applies u^ℓ as the feature for the final logistic regression classifier. However, ResNet-type models employ r^ℓ, which is in the dual space of u^ℓ in multigrid theory [18]. We provide the following three perspectives to understand the modified (pre-act) ResNet models. First, sharing A comes as a natural result of the connections between ResNet and MgNet under the framework of the constrained linear model, since there is only one linear system on
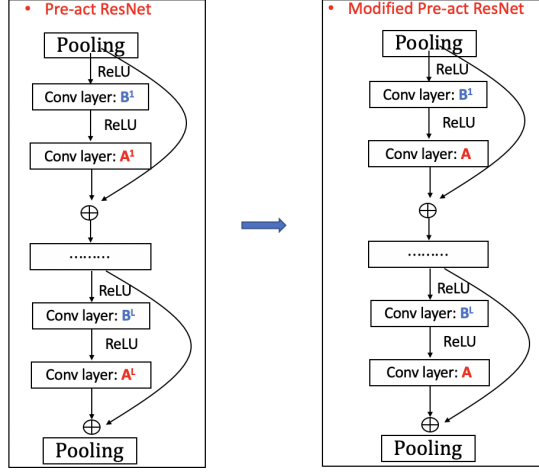
11

Figure 4.1: The architecture diagram of the modified pre-act ResNet. The key feature is to share the second convolutional kernels in basic blocks between two consecutive pooling layers.

each level. However, for any given linear system A^ℓ, applying different smoother $B^{\ell,i}$ at each residual correction step can improve the convergence of the iterative method; for example, the Chebyshev iteration and the multi-step iteration [48, 49] for solving linear systems. In addition, results in [50, 51] demonstrate that there is a high level of redundancy in (pre-act) ResNet models. Thus, sharing A at each level (change $A^{\ell,i}$ to A^ℓ) may not damage the representation power of (pre-act) ResNet models. Furthermore, we observe that A and B play different roles in the pre-act ResNet model in terms of their relations to the output. Recall the original basic block of pre-act ResNet

$$r^{\ell,i} = r^{\ell,i-1} + A^{\ell,i} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}) = r^{\ell,i-1} + A^{\ell,i} * \widetilde{B}^{\ell,i}(r^{\ell,i-1}),$$

where $\widetilde{B}^{\ell,i}(r) := \sigma \circ B^{\ell,i} * \sigma(r)$. Thus, we see that $r^{\ell,i}$ is linearly dependent on the kernel $A^{\ell,i}$, while $r^{\ell,i}$ is nonlinearly dependent on the kernel $B^{\ell,i}$. This simple observation indicates that sharing B may have completely different effects compared with sharing A. Numerical results in Table 3 further verify this observation.

A detailed numerical study of modified ResNet-type models in the following section in Tables 2 and 3 demonstrate the rationality of the constrained linear data-feature assumption that constitutes the foundation of these modified models.

4.3. Linear versus nonlinear data-feature mapping

In this subsection, we investigate the rationality of the linear assumption in data-feature mapping. We show that the linear data-feature-mapping model is both more reasonable and more accurate than general data-feature mapping iterative models.

12

4.3.1. A general data-feature iterative model

Two of the most important assumptions above are that data-feature mapping (4.1) is a linear model and that there should be only one model in each grid. To demonstrate that this linear model is adequate for image classification, we compare it with the following nonlinear data-feature mapping:

$$\mathcal{A}^\ell(u^\ell) = f^\ell, \tag{4.8}$$

where $\mathcal{A}^\ell(\cdot)$ can be chosen for some special nonlinear forms, such as $A^\ell * \sigma(\cdot)$, $\sigma \circ A^\ell * (\cdot)$, or $\sigma \circ A^\ell * \sigma(\cdot)$. Then, we have the next iterative feature-extraction scheme:

$$u^{\ell,i} = u^{\ell,i-1} + \mathcal{B}^{\ell,i}(f^\ell - \mathcal{A}^\ell(u^{\ell,i-1})), \ \ i = 1 : \nu_\ell, \tag{4.9}$$

where $\mathcal{B}^{\ell,i}$ takes linear or nonlinear forms. Here, we note that because of the nonlinearity of $\mathcal{A}^\ell$ we cannot obtain the iterative scheme for the residuals for (4.9). We can execute the iteration only in the feature space. Thus, we propose the next general data-feature iterative model (GDFI) in Algorithm 3, which follows a similar mechanism in MgNet, for example the iteration of features as in (4.9) and the pooling structure as in (4.11) and (4.12).

Algorithm 3 $u^{J,\ell_J} = \text{GDFI}(f; J, \nu_1, \cdots, \nu_J)$

1: Initialization: $f^1 = f_{\text{in}}(f)$, $u^{1,0} = 0$
2: **for** $\ell = 1 : J$ **do**
3:     **for** $i = 1 : \nu_\ell$ **do**
4:         Feature extraction (smoothing):

$$u^{\ell,i} = u^{\ell,i-1} + \mathcal{B}^{\ell,i}\left(f^\ell - \mathcal{A}^\ell(u^{\ell,i-1})\right). \tag{4.10}$$

5:     **end for**
6:     Pooling (interpolation and restriction):

$$u^{\ell+1,0} = \Pi_\ell^{\ell+1} *_2 u^{\ell,\nu_\ell}. \tag{4.11}$$

$$f^{\ell+1} = R_\ell^{\ell+1} *_2 \left(f^\ell - \mathcal{A}^\ell(u^{\ell,\nu_\ell}) + \mathcal{A}^{\ell+1}(u^{\ell+1,0})\right). \tag{4.12}$$

7: **end for**

Here, (4.11) and (4.12) are understood as different interpolation and restriction operators because of the difference in the feature and data space as discussed in relation to MgNet. However, in practice, all are implemented by 3×3 convolution with stride 2.

4.3.2. Numerical study indicating rationality of MgNet

If we take this specific setting,

$$
\begin{aligned}
\mathcal{A}^\ell(u) &= A^\ell * u, \\
\mathcal{B}^{\ell,i}(r) &= \sigma \circ B^{\ell,i} * \sigma(r),
\end{aligned}
\tag{4.13}
$$

then Algorithm 3 precisely degenerates to MgNet. The iterative scheme for its residual, therefore, becomes

$$
r^{\ell,i} = r^{\ell,i-1} - A^\ell * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}),
\tag{4.14}
$$

which is exactly the modified pre-act ResNet scheme discussed.

Considering the special form of MgNet, we try some numerical experiments with "symmetric" forms for different linear or nonlinear schemes for both $\mathcal{A}^\ell(\cdot)$ and $\mathcal{B}^{\ell,i}(\cdot)$ in Algorithm 3 as

$$
K * (\cdot), \;\; K * \sigma(\cdot), \;\; \sigma \circ K * (\cdot), \text{ and } \sigma \circ K * \sigma(\cdot),
\tag{4.15}
$$

where K is a 3×3 convolution kernel with multichannel, zero padding, and stride 1. Here, we recall that the key idea in developing pre-act ResNet [5] from ResNet [4] is to choose a better position for activation and convolution. Thus, from another perspective, the motivation for choosing $\mathcal{A}^\ell$ and $\mathcal{B}^{\ell,i}$ as in (4.15) is to study the dual version of the idea in developing pre-act ResNet in feature space.

As the results presented in Table 1 show, the original assumption about the linearity of $\mathcal{A}^\ell$ and the special non-linear form of $\mathcal{B}^{\ell,i}$, which forms MgNet exactly, is the most rational and accurate scheme. This result is also consistent with the theoretical concern and other numerical results in the following section.

5. Numerical experiments

In this section, we design numerical experiments to show that fixing the linear data-feature mapping in each produces only slightly negative or sometimes even positive effects as compared with the standard ResNet and pre-act ResNet models, which demonstrates the rationality of the constrained data-feature mapping model. Then, we compare the results of the MgNet model with those of established CNN models, and design a set of numerical experiments to explore the properties of MgNet and its variants.

Datasets.. We evaluate our various models using four widely used datasets: MNIST [2], CIFAR10 [46], CIFAR100 [46], and ImageNet(ILSVRC2012) [52].

Model implementation.. In our experiments, the structure of the classical ResNet and pre-act ResNet models is implemented with the same structure as in the sample codes in PyTorch [53]. We also implement our modified models and MgNet[1] with PyTorch. Following the strategy in [4, 5], we adopt Batch Normalization [54] but not Dropout [55].

Training.. We adopt the SGD training algorithm with momentum 0.9. We also adopt a weight decay value of 0.0005 on MNIST and CIFAR, whereas the value for ImageNet is 0.0001. We take the minibatch sizes to be 128, 128, 256 for MNIST, CIFAR, and ImageNet, respectively. We use the Kaiming's weight-initialization strategy as in [56]. We always start training with a learning rate of 0.1. For MNIST, we terminate training at 60 epochs and divide the learning rate by 10 at the 50-th epoch. For both CIFAR and ImageNet, we terminate training at 150 epochs and divide the learning rate by 10 at every 30 epochs.

5.1. Numerical results for modified ResNet and pre-act ResNet

To verify the optimality of the linear assumption of $\mathcal{A}^\ell$, we retain the linearity assumption of $\mathcal{A}^\ell$ with the iterative method (4.9). We, therefore, have the following iterative scheme for residuals $r^{\ell,i} = f^\ell - \mathcal{A}^\ell(u^{\ell,i})$:

$$r^{\ell,i} = r^{\ell,i-1} - \mathcal{A}^\ell \circ \mathcal{B}^{\ell,i}(r^{\ell,i-1}). \tag{5.1}$$

If we take the specific setting

$$
\begin{aligned}
\mathcal{A}^\ell(u) &= A^\ell * u, \\
\mathcal{B}^{\ell,i}(r) &= \sigma \circ B^{\ell,i} * \sigma(r),
\end{aligned}
\tag{5.2}
$$

the iterative scheme for the residuals (5.1) becomes

$$r^{\ell,i} = r^{\ell,i-1} - A^\ell * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}), \tag{5.3}$$

which is precisely the modified pre-act ResNet scheme (4.6). We compare it with some other linear forms and nonlinear forms for both $\mathcal{A}^\ell$ and $\mathcal{B}^{\ell,i}$ in Table 1, which shows that the modified pre-act ResNet scheme (4.6) is the most accurate. The result verifies that the assumption about the linearity of $\mathcal{A}^\ell$ and the special nonlinear form of $\mathcal{B}^{\ell,i}$ gives the most rational and accurate scheme, which is also consistent with the theoretical explanation in this paper.

---

[1]Codes are available at https://github.com/XuTeam/MgNet_Code.

Schemes of $\mathcal{A}^\ell$ and $\mathcal{B}^{\ell,i}$	Accuracy
$\mathcal{A}^\ell = A^\ell *,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} *$	71.36
$\mathcal{A}^\ell = A^\ell *,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} *$	93.04
$\mathcal{A}^\ell = A^\ell *,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} * \sigma$	93.80
$\mathcal{A}^\ell = A^\ell *,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} * \sigma$	**94.21**
$\mathcal{A}^\ell = A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} *$	92.90
$\mathcal{A}^\ell = A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} *$	92.87
$\mathcal{A}^\ell = A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} * \sigma$	94.01
$\mathcal{A}^\ell = A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} * \sigma$	93.98
$\mathcal{A}^\ell = \sigma \circ A^\ell *,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} *$	92.13
$\mathcal{A}^\ell = \sigma \circ A^\ell *,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} *$	92.44
$\mathcal{A}^\ell = \sigma \circ A^\ell *,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} * \sigma$	93.79
$\mathcal{A}^\ell = \sigma \circ A^\ell *,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} * \sigma$	93.57
$\mathcal{A}^\ell = \sigma \circ A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} *$	93.20
$\mathcal{A}^\ell = \sigma \circ A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} *$	93.93
$\mathcal{A}^\ell = \sigma \circ A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = B^{\ell,i} * \sigma$	94.08
$\mathcal{A}^\ell = \sigma \circ A^\ell * \sigma,\ \ \mathcal{B}^{\ell,i} = \sigma \circ B^{\ell,i} * \sigma$	94.11

Table 1: Accuracy of models from Algorithm 3 with different linear and non-linear schemes of $\mathcal{A}$ and $\mathcal{B}$ on CIFAR10.

Modified pre-act ResNet can also be understood as a special parameter-sharing form on $A^{\ell,i}$. To show that the effectiveness of the linear model does not arise from the redundancy of the CNN models, we also apply the parameter-sharing technique to $B^{\ell,i}$ for both ResNet and pre-act ResNet:

Pre-act ResNet-$A^{\ell,i}$-B^ℓ

$$r^{\ell,i} = r^{\ell,i-1} + A^{\ell,i} * \sigma \circ B^\ell * \sigma(r^{\ell,i-1}), \quad i = 1 : \nu_\ell. \tag{5.4}$$

Pre-act ResNet-A^ℓ-B^ℓ

$$r^{\ell,i} = r^{\ell,i-1} + A^\ell * \sigma \circ B^\ell * \sigma(r^{\ell,i-1}), \quad i = 1 : \nu_\ell. \tag{5.5}$$

As $B^{\ell,0}$ will change the channel number in ResNet (3.2) and pre-act ResNet (3.4), we share only $B^{\ell,i}$ for $i = 1 : \nu_\ell$. For consistency, we denote the original ResNet and pre-act ResNet models as ResNet-$A^{\ell,i}$-$B^{\ell,i}$ and pre-act ResNet-$A^{\ell,i}$-$B^{\ell,i}$, respectively.

Model	Accuracy	# Parameters
ResNet18-$A^{\ell,i}$-$B^{\ell,i}$	99.49	11M
ResNet18-A^{ℓ}-$B^{\ell,i}$	**99.61**	8.0M
pre-act ResNet18-$A^{\ell,i}$-$B^{\ell,i}$	99.63	11M
pre-act ResNet18-A^{ℓ}-$B^{\ell,i}$	**99.67**	8.0M

Table 2: Accuracy and number of parameters of ResNet-18, pre-act ResNet-18, and their modified models on MNIST.

Model	CIFAR10	CIFAR100	# Parameters
ResNet18-$A^{\ell,i}$-$B^{\ell,i}$	94.22	76.08	11M
ResNet18-A^{ℓ}-$B^{\ell,i}$	**94.34**	**76.32**	8.1M
ResNet18-$A^{\ell,i}$-B^{ℓ}	93.95	74.23	9.7M
ResNet18-A^{ℓ}-B^{ℓ}	93.30	74.85	6.6M
pre-act ResNet18-$A^{\ell,i}$-$B^{\ell,i}$	94.31	76.33	11M
pre-act ResNet18-A^{ℓ}-$B^{\ell,i}$	**94.54**	**76.43**	8.1M
pre-act ResNet18-$A^{\ell,i}$-B^{ℓ}	93.96	74.45	9.7M
pre-act ResNet18-A^{ℓ}-B^{ℓ}	93.63	74.46	6.6M
ResNet34-$A^{\ell,i}$-$B^{\ell,i}$	94.43	76.31	21M
ResNet34-A^{ℓ}-$B^{\ell,i}$	**94.78**	**76.44**	13M
ResNet34-$A^{\ell,i}$-B^{ℓ}	93.98	74.48	15M
ResNet34-A^{ℓ}-B^{ℓ}	93.55	74.46	6.7M
pre-act ResNet34-$A^{\ell,i}$-$B^{\ell,i}$	94.70	77.38	21M
pre-act ResNet34-A^{ℓ}-$B^{\ell,i}$	**94.91**	**77.41**	13M
pre-act ResNet34-$A^{\ell,i}$-B^{ℓ}	94.08	75.32	15M
pre-act ResNet34-A^{ℓ}-B^{ℓ}	94.01	74.12	6.7M

Table 3: Accuracy and number of parameters for ResNet, pre-act ResNet, and their variants of modified versions on CIFAR10 and CIFAR100.

Importantly, in relation to the numerical results shown in Table 2 and Table 3, the modified ResNet and pre-act ResNet models achieve almost the same accuracy as their original respective models, whereas the other models do not. This result indicates that the constrained data-feature mapping properly captures the mathematical insight of the ResNet-related models.

5.2. MgNet vs. established neural networks

According to the discussion and numerical results in §4.3, MgNet is the most natural and accurate model under the general data-feature iterative scheme. In the following subsections, we present a systematic numerical study to demonstrate the success of MgNet for image classification problems and, in this context, its advantages over established networks.

First, we test MgNet on the CIFAR10, CIFAR100, and ImageNet datasets and compare the results with AlexNet [3], VGG [7], ResNet [4], pre-act ResNet [5], and WideResNet [10]. As shown in Table 5.2, MgNet achieves 96% accuracy on CIFAR10 and 79.94% accuracy on CIFAR100. On the ImageNet dataset, MgNet achieves 78.59% top-1 accuracy. Compared with the selected benchmark models, MgNet, therefore, is more accurate and has fewer parameters which demonstrate its superior effectiveness as compared with the other models.

Dataset	Model	Accuracy	Parameters
CIFAR10	AlexNet [3][a]	76.22	2.5M
	VGG19 [7] [a]	93.56	20.0M
	ResNet18 [4]	95.28	11.2M
	pre-act ResNet1001 [5]	95.08	10.2M
	WideResNet28 $*$ 2 [10]	95.83	36.5M
	MgNet[2,2,2,2]-256-B^l	**96.00**	8.2M
CIFAR100	AlexNet [3] [a]	43.87	2.5M
	VGG19 [7] [a]	71.95	20.0M
	ResNet18 [4]	77.54	11.2M
	preact-ResNet1001 [5]	77.29	10.2M
	WideResNet40 $*$ 2 [10]	79.50	36.5M
	MgNet[2,2,2,2]-256-B^l	**79.94**	8.3M
ImageNet	AlexNet [3]	63.30	60.2M
	VGG19 [7]	74.50	144.0M
	ResNet18 [4]	72.12	11.2M
	preact-ResNet200 [5]	78.34	64.7M
	WideResNet50 $*$ 2 [10]	78.10	68.9M
	MgNet[3,4,6,3]-[128,256,512,1024]-$B^{l,i}$	**78.59**	71.3M

[a]Results are reported in https://reposhub.com/python/deep-learning/bearpaw-pytorch-classification.html

Table 4: Accuracy of MgNet and established CNN models for widely used datasets.

5.3. MgNet with different channels

Next, we employ two MgNet variants to demonstrate the model's scalability with respect to the number of channels. The first version is consistent with the typical CNN models; as the grid becomes deeper, the number of channels gradually increases. In the second version, the number of channels is the same across the grids, i.e., the number of channels does not change with the grids (resolution). For all cases on CIFAR100 (Table 5), accuracy improves simultaneously with the number of parameters. We also found that on CIFAR100, with the same number of parameters, MgNet with fixed channels is more accurate than MgNet with increasing channels, as shown in Table 5. Based on this fact, the number of channels for MgNet in relation to the CIFAR datasets is fixed in each grid in the rest of this paper. On ImageNet, as MgNet with fixed channels has a huge number of parameters, we adopt MgNet with increasing channels in the rest of the paper. As shown in Table 6, accuracy improves simultaneously on ImageNet as the number of parameters increases. These results show that MgNet has great potential for scalability in terms of the number of channels.

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
CIFAR100	MgNet[2,2,2,2]-[256]-B^l	79.94	8.3M
	MgNet[2,2,2,2]-[512]-B^l	81.35	33.1M
	MgNet[2,2,2,2]-[768]-B^l	81.74	74.4M
	MgNet[2,2,2,2]-[1024]-B^l	81.89	132.2M
	MgNet[2,2,2,2]-[32,64,128,256]-B^l	74.95	2.3M
	MgNet[2,2,2,2]-[64,128,256,512]-B^l	78.06	12.5M
	MgNet[2,2,2,2]-[128,256,512,1024]-B^l	80.29	37.5M
	MgNet[2,2,2,2]-[256,512,1024,2048]-B^l	81.49	150.0M

Table 5: MgNet with fixed and increasing channels on CIFAR100

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
ImageNet	MgNet[2,2,2,2]-[64,128,256,512]-B^ℓ	72.32	9.9M
	MgNet[2,2,2,2]-[128,256,512,1024]-B^ℓ	76.82	38.5M

Table 6: MgNet with increasing number of channels on ImageNet.

5.4. MgNet with different number of iterations ν_ℓ

In this subsection, we explore the impact of the number of iterations ν_ℓ of each grid in MgNet. We change only ν_ℓ in the grids and keep all the other parameters fixed. In Table 7, we can see that as ν_1 increases, the corresponding accuracy improves. We also perform similar tests on the other layers, except for the first grid; increasing the number of iterations of the other grids, ν_2, ν_3, ν_4, has no significant impact on the accuracy of the model, as shown in Table 8. In addition, we test different ν_3 values on ImageNet and find that as ν_3 increases the corresponding accuracy improves, as shown in Table 9. Increasing the number of iterations ν_ℓ does not increase the number of parameters, which is also an advantage that MgNet has over the other models tested.

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
CIFAR100	MgNet[2,2,2,2]-[256]-B^ℓ	79.94	8.3M
	MgNet[4,2,2,2]-[256]-B^ℓ	80.25	8.3M
	MgNet[8,2,2,2]-[256]-B^ℓ	80.32	8.3M
	MgNet[16,2,2,2]-[256]-B^ℓ	80.42	8.3M
	MgNet[32,2,2,2]-[256]-B^ℓ	80.89	8.3M
	MgNet[2,2,2,2]-[512]-B^ℓ	81.35	33.1M
	MgNet[4,2,2,2]-[512]-B^ℓ	81.53	33.1M
	MgNet[8,2,2,2]-[512]-B^ℓ	81.83	33.1M
	MgNet[16,2,2,2]-[512]-B^ℓ	81.97	33.1M
	MgNet[2,2,2,2]-[1024]-B^ℓ	81.89	132.2M
	MgNet[8,2,2,2]-[1024]-B^ℓ	82.46	132.2M

Table 7: MgNet with different ν_1 on CIFAR100.

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
	MgNet[2,2,2,2]-[256]-B^ℓ	79.94	8.3M
	MgNet[2,4,2,2]-[256]-B^ℓ	79.96	8.3M
	MgNet[2,8,2,2]-[256]-B^ℓ	79.92	8.3M
	MgNet[2,16,2,2]-[256]-B^ℓ	79.97	8.3M
	MgNet[2,2,2,2]-[256]-B^ℓ	79.94	8.3M
	MgNet[2,2,4,2]-[256]-B^ℓ	79.85	8.3M
CIFAR100	MgNet[2,2,8,2]-[256]-B^ℓ	79.91	8.3M
	MgNet[2,2,16,2]-[256]-B^ℓ	79.77	8.3M
	MgNet[2,2,2,2]-[256]-B^ℓ	79.94	8.3M
	MgNet[2,2,2,4]-[256]-B^ℓ	79.60	8.3M
	MgNet[2,2,2,8]-[256]-B^ℓ	79.28	8.3M
	MgNet[2,2,2,16]-[256]-B^ℓ	79.47	8.3M

Table 8: MgNet with different ν_2, ν_3, ν_4 on CIFAR100.

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
	MgNet[2,2,2,2]-[64,128,256,512]-B^ℓ	72.32	9.9M
ImageNet	MgNet[2,2,4,2]-[64,128,256,512]-B^ℓ	73.04	9.9M
	MgNet[2,2,8,2]-[64,128,256,512]-B^ℓ	73.72	9.9M
	MgNet[2,2,16,2]-[64,128,256,512]-B^ℓ	73.81	9.9M

Table 9: MgNet with different ν_3 on ImageNet.

5.5. Parameter sharing on operator B

Here, we explore the parameter-sharing technique on operator B. We consider two cases: B^ℓ (B operator of every iteration step in the same grid is the same) and $B^{\ell,i}$ (B operators of every iteration step in the same grid are different). The influence of B is tested on CIFAR100 and ImageNet. As shown in Table 10 and Table 11, the MgNet-$B^{\ell,i}$ models have a larger number of parameters and are more accurate compared with the MgNet-B^ℓ models.

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
	MgNet[2,2,2,2]-[256]-B^ℓ	79.94	8.3 M
	MgNet[2,2,2,2]-[256]-$B^{\ell,i}$	80.12	10.7M
CIFAR100	MgNet[4,2,2,2]-[256]-B^ℓ	80.25	8.3 M
	MgNet[4,2,2,2]-[256]-$B^{\ell,i}$	80.63	11.9M
	MgNet[8,2,2,2]-[256]-B^ℓ	80.32	8.3M
	MgNet[8,2,2,2]-[256]-$B^{\ell,i}$	81.42	14.3M

Table 10: Comparison of MgNet with share B and no share B on CIFAR100.

Dataset	$[\nu_1, \nu_2, \cdots, \nu_J], c_\ell$	Accuracy	Parameters
	MgNet[2,2,2,2]-[64,128,256,512]-B^ℓ	72.32	9.9M
	MgNet[2,2,2,2]-[64,128,256,512]-$B^{\ell,i}$	73.36	13.0M
	MgNet[2,2,4,2]-[64,128,256,512]-B^ℓ	73.04	9.9M
ImageNet	MgNet[2,2,4,2]-[64,128,256,512]-$B^{\ell,i}$	74.58	14.7M
	MgNet[2,2,2,2]-[128,256,512,1024]-B^ℓ	76.82	38.5M
	MgNet[2,2,2,2]-[128,256,512,1024]-$B^{\ell,i}$	77.27	51.1M
	MgNet[2,2,4,2]-[128,256,512,1024]-B^ℓ	77.58	38.5M
	MgNet[2,2,4,2]-[128,256,512,1024]-$B^{\ell,i}$	77.94	55.7M

Table 11: Comparison of MgNet with share B and no share B on ImageNet.

6. Discussion and conclusion

We proposed a constrained linear data-feature-mapping model as underlying ResNet and MgNet to demonstrate their dual relation. Under this model, we investigated the connections between the traditional iterative method with a nonlinear constraint and the basic block scheme in the pre-act ResNet model, and developed an explanation for pre-act ResNet at a technical level from the dual perspective of MgNet. In comparison with existing studies that discuss the connection between dynamic systems and ResNet, the constrained data-feature-mapping model goes beyond both formal and qualitative comparisons to identify key model components via a more detailed account. Furthermore, we hope that the reason, and the ways in which, ResNet-type models work can be mathematically understood in a similar fashion as is the case for classical iterative methods in scientific computing for which the theoretical understanding is more mature and better-developed. The numerical experiments verified in this paper indicate the rationality and efficiency for the con-

strained learning data-feature-mapping model. In addition, a systematic numerical study on MgNet shows its success in image classification problems and its advantages over established networks.

We believe that our investigation into the connections between CNNs and classical iterative methods opens a new door to the mathematical understanding and analysis of CNN models with certain structures as well as creating opportunities to make improvements to them. The results presented indicate the great potential of this model from both theoretical and empirical viewpoints. Obviously many aspects of classical iterative methods with constraint should be further explored with the goal of making significant improvements in this regard. For example, we are currently focusing on establishing the connection of ResNet with bottleneck and the subspace correction iterative methods [16] and applying different techniques from iterative methods to MgNet.

Acknowledgements

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.

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

[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[5] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, 2016, pp. 630–645.

[6] J. He, J. Xu, Mgnet: A unified framework of multigrid and convolutional neural network, Science China Mathematics (2019) 1–24.

[7] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[9] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[10] S. Zagoruyko, N. Komodakis, Wide residual networks, in: British Machine Vision Conference, 2016, pp. 87.1–87.12.

[11] G. Larsson, M. Maire, G. Shakhnarovich, Fractalnet: Ultra-deep neural networks without residuals, arXiv preprint arXiv:1605.07648 (2016).

[12] A. N. Gomez, M. Ren, R. Urtasun, R. B. Grosse, The reversible residual network: Backpropagation without storing activations, in: Advances in Neural Information Processing Systems, 2017, pp. 2214–2224.

[13] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 5987–5995.

[14] X. Zhang, Z. Li, C. C. Loy, D. Lin, Polynet: A pursuit of structural diversity in very deep networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 3900–3908.

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[16] J. Xu, Iterative methods by space decomposition and subspace correction, SIAM Review 34 (1992) 581–613.

[17] W. Hackbusch, Multi-grid Methods and Applications, volume 4, Springer Science & Business Media, 2013.

[18] J. Xu, L. Zikatanov, Algebraic multigrid methods, Acta Numerica 26 (2017) 591–721.

[19] J. Xu, L. Zikatanov, The method of alternating projections and the method of subspace corrections in Hilbert space, Journal of the American Mathematical Society 15 (2002) 573–597.

[20] T.-W. Ke, M. Maire, X. Y. Stella, Multigrid neural architectures, arXiv preprint arXiv:1611.07661 (2016).

[21] E. Haber, L. Ruthotto, E. Holtham, Learning across scales-a multiscale method for convolution neural networks, arXiv preprint arXiv:1703.02009 (2017).

[22] L. Zhang, Z. Tan, J. Song, J. Chen, C. Bao, K. Ma, Scan: A scalable neural networks framework towards compact and efficient models, arXiv preprint arXiv:1906.03951 (2019).

[23] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[24] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 fourth international conference on 3D vision (3DV), IEEE, 2016, pp. 565–571.

[25] A. Katrutsa, T. Daulbaev, I. Oseledets, Deep multigrid: learning prolongation and restriction matrices, arXiv preprint arXiv:1711.03825 (2017).

[26] J.-T. Hsieh, S. Zhao, S. Eismann, L. Mirabella, S. Ermon, Learning neural pde solvers with convergence guarantees, ICLR 2019 (2019).

[27] W. E, A proposal on machine learning via dynamical systems, Communications in Mathematics and Statistics 5 (2017) 1–11.

[28] B. Chang, L. Meng, E. Haber, F. Tung, D. Begert, Multi-level residual networks from dynamical systems view, arXiv preprint arXiv:1710.10348 (2017).

[29] Y. Lu, A. Zhong, Q. Li, B. Dong, Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations, in: Proceedings of the 35th International Conference on Machine Learning, volume 80, PMLR, 2018.

[30] T. Q. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, Neural ordinary differential equations, in: Advances in neural information processing systems, 2018, pp. 6571–6583.

[31] S. Sonoda, N. Murata, Double continuum limit of deep neural networks, in: ICML Workshop Principled Approaches to Deep Learning, 2017.

[32] Z. Li, Z. Shi, A flow model of neural networks, arXiv preprint arXiv:1708.06257v2 (2017).

[33] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the 27th International Conference on Machine Learning, Omnipress, 2010, pp. 399–406.

[34] J. Sun, H. Li, Z. Xu, et al., Deep admm-net for compressive sensing mri, in: Advances in Neural Information Processing Systems, 2016, pp. 10–18.

[35] H. Li, Y. Yang, D. Chen, Z. Lin, Optimization algorithm inspired deep neural network structure design, arXiv preprint arXiv:1810.01638 (2018).

[36] U. Helmke, J. B. Moore, Optimization and dynamical systems, Springer Science & Business Media, 2012.

[37] A. Veit, M. J. Wilber, S. Belongie, Residual networks behave like ensembles of relatively shallow networks, Advances in neural information processing systems 29 (2016).

[38] E. Littwin, L. Wolf, The loss surface of residual networks: Ensembles and the role of batch normalization, arXiv preprint arXiv:1611.02525 (2016).

[39] F. Huang, J. Ash, J. Langford, R. Schapire, Learning deep resnet blocks sequentially using boosting theory, in: International Conference on Machine Learning, PMLR, 2018, pp. 2058–2067.

[40] A. Nitanda, T. Suzuki, Functional gradient boosting based on residual network perception, in: International Conference on Machine Learning, PMLR, 2018, pp. 3819–3828.

[41] Z. Allen-Zhu, Y. Li, What can resnet learn efficiently, going beyond kernels?, Advances in Neural Information Processing Systems 32 (2019).

[42] K. Huang, Y. Wang, M. Tao, T. Zhao, Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective, Advances in neural information processing systems 33 (2020) 2698–2709.

[43] T. Tirer, J. Bruna, R. Giryes, Kernel-based smoothness analysis of residual networks, in: Mathematical and Scientific Machine Learning, PMLR, 2022, pp. 921–954.

[44] H. Lin, S. Jegelka, Resnet with one-neuron hidden layers is a universal approximator, Advances in neural information processing systems 31 (2018).

[45] J. He, L. Li, J. Xu, Approximation properties of deep relu cnns, Research in the Mathematical Sciences 9 (2022) 38.

[46] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, IEEE, 2009, pp. 248–255.

[48] W. Hackbusch, Iterative Solution of Large Sparse Systems of Equations, volume 95, Springer, 1994.

[49] G. H. Golub, C. F. Van Loan, Matrix Computations, volume 3, JHU Press, 2012.

[50] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, International Conference on Learning Representations (2015).

[51] J. He, X. Jia, J. Xu, L. Zhang, L. Zhao, Make ℓ_1 regularization effective in training sparse cnn, Computational Optimization and Applications 77 (2020) 163–182.

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015) 211–252.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito,

M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8026–8037.

[54] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (2014) 1929–1958.

[56] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International conference on Computer Vision, 2015, pp. 1026–1034.