The 27th International Conference on Neural Information Processing (ICONIP2020), Nov 18–22, 2020

Why do Deep Neural Networks with Skip Connections and Concatenated Hidden Representations Work?

Oyebade K. Oyedotun ✉ and Djamila Aouada

Interdisciplinary Centre for Security, Reliability and Trust (SnT),
University of Luxembourg, L-1855 Luxembourg
{oyebade.oyedotun,djamila.aouada}@uni.lu

http://wwwen.uni.lu/snt

Abstract. Training the classical-vanilla deep neural networks (DNNs) with several layers is problematic due to optimization problems. Interestingly, skip connections of various forms (e.g. that perform the summation or concatenation of hidden representations or layer outputs) have been shown to allow the successful training of very DNNs. Although there are ongoing theoretical works to understand very DNNs that employ the summation of the outputs of different layers (e.g. as in the residual network), there is none to the best of our knowledge that has studied why DNNs that concatenate of the outputs of different layers (e.g. as seen in Inception, FractalNet and DenseNet) works. As such, we present in this paper, the first theoretical analysis of very DNNs with concatenated hidden representations based on a general framework that can be extended to specific cases. Our results reveal that DNNs with concatenated hidden representations circumnavigate the singularity of hidden representation, which is catastrophic for optimization. For substantiating the theoretical results, extensive experiments are reported on standard datasets such as the MNIST and CIFAR-10.

Keywords: Deep networks, skip connection, optimization, generalization

1 Introduction

Classical deep neural networks (DNNs) have only one path from the input layer to the output layer for information flow. This type of DNNs are commonly referred to as 'PlainNets', and many works [1,2] have reported improved results on various tasks by simply extending the depth of previous state-of-the-art PlainNets. Following this observation, theoretical studies that characterize the role of model depth for the compact representation of complicated functions can be found in [3,4]. Interestingly, training PlainNets with many layers of feature representations (i.e. very deep PlainNets) is difficult, since optimization typically becomes problematic when the number of model layers exceeds fifteen [5,6]. However, it is known that the optimization problem of very deep PlainNets can be mitigated by employing skip connections of various forms. A popular form of skip connection in the literature is based on the summation of the output of any given layer with the outputs of earlier layers. Some very DNNs that adopt this type of construction include the ResNet [7], ResNeXt [8] and S-ResNet [9]. While

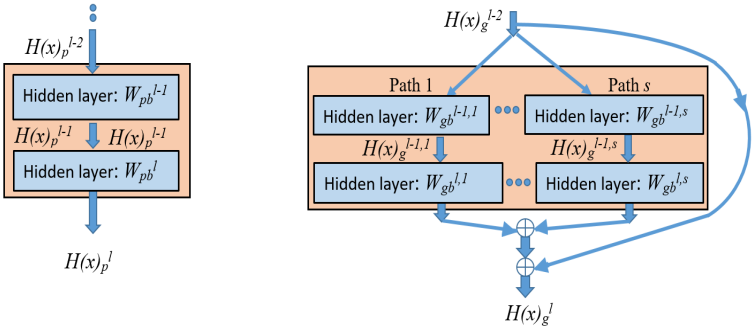

Fig. 1: Generic DNN block used for analysis, where $\oplus$ denotes the vertical concatenation operation. Left: PlainNet block. Right: ConcNeXt block.

very impressive results for different tasks have been reported in several works using skip connections, the literature is still lacking of a concrete theoretical account of their operation. We note that analytical studies have been limited to the ResNet, since using skip connections that sum the hidden representation of different layers generally results in complicated DNN models. Notwithstanding, controversies such as the useful number of layers [10], how optimization problem is circumnavigated [10,11,12], and the source of improved generalization capacity [10] continue to trail the ResNet. Hence, it is not surprising that the theoretical analysis of similar but more complicated models such as the ResNeXt [8] and S-ResNet [9] are outrightly missing in the literature.

The other popular and successful form of skip connection in the literature entails the concatenation of the output of different layers. DNNs that employ this type of construction are the Inception [13], Inception V3 [14], FractalNet [15] and DenseNet [16]. Despite the success of these DNN models, to the best of our knowledge, there are no theoretical works to understand their operation. This is not surprising, given the complicated architectural construction of this type of DNNs in comparison to the DNNs that employ skip connections and the summation of hidden representations [7,8,9].

In this paper, we provide a theoretical framework for understanding the complicated operation of very DNN models that employ skip connections with concatenated hidden representations. We acknowledge the proliferation of DNN architectures that employ skip connections and concatenated hidden representations in the literature [13,14,15,16]. As such, a generic DNN architecture that can be adapted to specific cases is employed for analysis in this paper. Specifically, our generic architecture is similar to the ResNeXt, where the summation operation is simply replaced by the concatenation operation. We borrow the nomenclature of the ResNeXt so that the DNN architecture used in this paper is referred to as 'ConcNeXt'. We note that the theoretical results for the generic DNN architecture used in this paper are relatable to similar DNNs such as the Inception [13] and DenseNet [16]. Namely, our contributions are as follows.

1. Present the first theoretical study of DNNs with concatenated hidden representations that relies on elements of linear algebra and random matrix theory. Our results unravel unique model characteristics that translate to a good optimization condition for models with several hidden layers.

2. Report extensive experimental results to support the analytical results using benchmarking datasets such as the MNIST and CIFAR-10.

The paper is organized as follows. Section 2 discusses related works. Section 3 presents the preliminaries on DNNs with skip connection and concatenated hidden representations using a generic DNN architecture. Theoretical analysis is given in Section 4. Experimental results with discussions are in Section 5. Section 6 concludes the paper.

2 Related Work

Deep neural networks with skip connections: Considering the proliferation of different DNN architectures, there is a growing concern in recent times of their interpretability. Particularly, we are interested in how the architectures of the DNNs impact their training characteristics and performances. Interestingly, the unconventional construction of the state-of-the-art DNNs, which employ various forms of skip connections obfuscate their operation. Nevertheless, we note that considerable progress has been made in understanding the operation of DNNs that employ the summation of the outputs of hidden layers with the outputs of the previous layers; particularly, the ResNet [7]. For instance, the work [10] argued that the effective depth of the ResNet is significantly smaller than the architectural depth. Namely, the ResNet with 110 layers was shown to have an effective depth of 17 layers [10]. The gradients shattering problem was studied in [11], where skip connections were found to be helpful for fostering well-structured gradients that make the training of very DNNs successful. The unrolled iterative estimation concept was proposed in [12,17], where it was argued that groups of ResNet blocks iteratively refine the representations computed in a particular stage, and new representations are computed in other stages [12]. Another work from the dynamical systems perspective is in [18]. Importantly, a unanimous account of the operation of the ResNet remains a challenge in the deep learning community. Subsequently, the literature is lacking of the theoretical study of DNNs that employ skip connections and the concatenation of the outputs of hidden layers with the outputs of previous layers. Some DNNs that use this type of construction include the Inception [13,14], FractalNet [15] and DenseNet [16]. We presume that the main reason for the lack of analytical study for this type of DNNs is their more complicated constructions and operations.

Studying deep linear neural networks: The strict theoretical study of DNNs with all its 'bells and whistles' often results in mathematical intractability. Among other simplifications that make DNN amenable to theoretical analysis is the linear activation function assumption [19,20,21]. In fact, stricter assumptions such as the number of hidden units, data points and convexity are in the literature [20,22]. Interestingly, it is known that the theoretical results obtained using linear DNNs are mostly applicable for non-linear DNNs [23,24]. This observation is not confounding, since a linear DNN with two or more layers is still a non-convex optimization problem in the parameter space.

3 Preliminaries

We discuss as preliminaries the simplified and generic DNN architecture used for the theoretical analysis in the paper. Specifically, the building blocks for the PlainNet and

ConcNeXt that the analyses are based on are shown in Fig. 1, where we use $\oplus$ to denote the vertical concatenation operation. Note that we assume the linear activation function for analysis. However, we show via experiments (in Section 5) that theoretical results are valid for practical models that use the non-linear activation function such as the rectified linear function. First, we give the following axiom that is important for expressing the hidden representations of the ConcNeXt in interesting forms.

Axiom 1. *Let a matrix $\boldsymbol{C} \in \mathbb{R}^{n \times N}$ be the vertical concatenation of matrices $\boldsymbol{A} \in \mathbb{R}^{n_1 \times N}$ and $\boldsymbol{B} \in \mathbb{R}^{n_2 \times N}$ as in $\boldsymbol{C} = (A \oplus B)$; where $n = n_1 + n_2$. Assuming $\boldsymbol{A} = \boldsymbol{DB}$ with $\boldsymbol{D} \in \mathbb{R}^{n_1 \times n_2}$, so that $\boldsymbol{C} = (\boldsymbol{DB} \oplus \boldsymbol{B})$. Then, we can write $\boldsymbol{C} = (\boldsymbol{D} \oplus \mathbf{I})\boldsymbol{B}$, where $\mathbf{I} \in \mathbb{R}^{n_2 \times n_2}$ is the identity matrix.*

3.1 Plain network (PlainNet)

The output of the PlainNet block is strictly hierarchical as in Fig. 1 (left); there is only one connecting input. Let the input data to the PlainNet DNN model be $\boldsymbol{X} \in \mathbb{R}^{n \times N}$. Subsequently, let the input to a hypothetical PlainNet block composed of two hidden layers be $\boldsymbol{H}(\boldsymbol{X})_p^{l-2} \in \mathbb{R}^{n \times N}$ as in Fig. 1 (left); where $\boldsymbol{W}_{pb}^{l} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{W}_{pb}^{l-1} \in \mathbb{R}^{n \times n}$ are the layers' parameters initialized as Gaussian random matrices, and pb indicates the weights in a PlainNet block. Thus, the output of the PlainNet block is

$$\boldsymbol{H}(\boldsymbol{X})_p^{l} = W_{pb}^{l} W_{pb}^{l-1} \boldsymbol{H}(\boldsymbol{X})_p^{l-2}. \tag{1}$$

For the sake of simplicity, the transformation $\boldsymbol{W}_{pb}^{l}\boldsymbol{W}_{pb}^{l-1}$ given in Eqn. (1) is lumped as $\boldsymbol{W}_{p}^{l} = \boldsymbol{W}_{pb}^{l}\boldsymbol{W}_{pb}^{l-1}$, where $\boldsymbol{W}_{p}^{l} \in \mathbb{R}^{n \times n}$. Hence, the output of the PlainNet block can be simply expressed as

$$\boldsymbol{H}(\boldsymbol{X})_p^{l} = \boldsymbol{W}_{p}^{l} \boldsymbol{H}(\boldsymbol{X})_p^{l-2}. \tag{2}$$

3.2 Concatened Network of hidden representations (ConcNeXt)

The ConcNeXt employs skip connections that concatenate the output of the previous block with the outputs of the s parallel paths in the current block in Fig. 1 (right); where $s \in \mathbb{N}^{+}$ is referred to as the cardinality of the ConcNeXt with $1 \leq k \leq s$. As such, let the the weight matrix at layer l and path k be $\boldsymbol{W}_{gb}^{l,k} \in \mathbb{R}^{q \times q}$, where gb indicates the weights in the ConcNeXt block. Similar to the ResNeXt [8], the parameterization of the ConcNeXt is such that $q \leftarrow q/s$. That is, the dimensions of the weight matrix is reduced with an increase in cardinality. For compactness, we use $\curlyvee_{k=1}^{s} \boldsymbol{\Gamma}^{k}$ to denote successive vertical concatenations of the variable $\boldsymbol{\Gamma}^{k}$, where the dimension of $\boldsymbol{\Gamma}^{k}$ permits the concatenation operations. That is, $\curlyvee_{k=1}^{s} \boldsymbol{\Gamma}^{k} = (\boldsymbol{\Gamma}^{1} \oplus \cdots \oplus \boldsymbol{\Gamma}^{k} \oplus \cdots \oplus \boldsymbol{\Gamma}^{s})$. Given the block's input, $\boldsymbol{H}(\boldsymbol{X})_g^{l-2} \in \mathbb{R}^{q \times N}$, the output of the ConcNeXt block, $\boldsymbol{H}(\boldsymbol{X})_g^{l}$, with the parameters $\boldsymbol{W}_{gb}^{l,k}, \boldsymbol{W}_{gb}^{l-1,k} \in \mathbb{R}^{q \times q}$ initialized as Gaussian random matrices is

$$\boldsymbol{H}(\boldsymbol{X})_g^{l} = (\curlyvee_{k=1}^{s} \boldsymbol{W}_{gb}^{l,k} \boldsymbol{W}_{gb}^{l-1,k} \boldsymbol{H}(\boldsymbol{X})_g^{l-2} \oplus \boldsymbol{H}(\boldsymbol{X})_g^{l-2}), \tag{3}$$

Again, for simplicity, the transformation $\boldsymbol{W}_{gb}^{l,k}\boldsymbol{W}_{gb}^{l-1,k}$ in Eqn. (3) is lumped as $\boldsymbol{W}_{g}^{l,k} = \boldsymbol{W}_{gb}^{l,k}\boldsymbol{W}_{gb}^{l-1,k}$, where $\boldsymbol{W}_{g}^{l,k} \in \mathbb{R}^{q \times q}$. Finally, factorizing Eqn. (3) using Axiom 1 gives

$$\boldsymbol{H}(\boldsymbol{X})_g^{l} = (\curlyvee_{k=1}^{s} \boldsymbol{W}_{g}^{l,k} \oplus \mathbf{I}) \boldsymbol{H}(\boldsymbol{X})_g^{l-2}. \tag{4}$$

4 Theoretical Study of DNNs with Skip Connections and Concatenated Hidden Represenations

This section theoretically investigates why DNNs with skip connections and concatenated hidden representations alleviate the training problems of very DNNs. The analysis of the ConcNeXt is positioned relative to the optimization problem of very deep PlainNets based on the singularity of hidden representations. First, the definitions, propositions and lemmas that are crucial for the theoretical study are given as follows. Note that all proofs are in the supplementary material.

Definition 1. *The condition number of a matrix* $\boldsymbol{A} \in \mathbb{R}^{n \times N}$ *denoted* $\kappa(\boldsymbol{A})$ *is given as*

$$\kappa(\boldsymbol{A}) = \sigma_{max}(\boldsymbol{A}) / \sigma_{min}(\boldsymbol{A}), \tag{5}$$

where $\sigma_{max}(\boldsymbol{A})$ and $\sigma_{min}(\boldsymbol{A})$ are the maximum and minimum singular values of $\boldsymbol{A}$, respectively. From Definition 1, singularity implies that $\sigma_{min}(\boldsymbol{A}) = 0$, so that $\kappa(\boldsymbol{A}) = \infty$. Generally, the optimization of ill-posed problems are very difficult [25, 26].

Proposition 1. *Given a matrix* $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ *whose column vectors,* $\boldsymbol{w}_i$*, are drawn from a Gaussian or uniform distribution, the probability that* $\boldsymbol{W}$ *is non-singular,* $P(\boldsymbol{w_i} \notin W_{-i}^{span})$*, is*

$$P(\boldsymbol{w}_i \notin W_{-i}^{span}) = 1 \ : 1 \leq i \leq n \tag{6}$$

Corollary 1. *The initialization [27, 28] of an* m*-layer DNN weight matrices,* $\{\boldsymbol{W}^l \in \mathbb{R}^{n \times n}\}_{l=1}^{m}$*, follows Proposition 1, and hence are all non-singular. That is,* $\sigma_{min}(\boldsymbol{W}^l) \neq 0 : 1 \leq l \leq m$*.*

Corollary 2. *For an* m*-layer DNN, Corrollary 1 gives* $0 < \sigma_{min}(\boldsymbol{W}^l) < \infty : 1 \leq l \leq m$*. Subsequently, popular weights initialization schemes [27, 28] yield* $P(\sigma_{min}(\boldsymbol{W}^l) < 1) = 1 : 1 \leq l \leq m$*.*

Assumption 1. *The input to an* m*-layer DNN,* $\boldsymbol{X} \in \mathbb{R}^{n \times N}$*, is non-singular. This is an important assumption required for the validity of many machine learning algorithms.*

Lemma 1. *Let* $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ *and* $\boldsymbol{V} \in \mathbb{R}^{n \times p}$ *be matrices. If* $\boldsymbol{V}$ *is singular, then their product,* $\boldsymbol{Y} = \boldsymbol{XV}$*, is also singular.*

Lemma 2. *Let* $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ *and* $\boldsymbol{V} \in \mathbb{R}^{n \times p}$ *be two matrices. If* $\boldsymbol{X}$ *and* $\boldsymbol{V}$ *are non-singular, then their product,* $\boldsymbol{Y} = \boldsymbol{XV}$*, is also non-singular.*

Lemma 3. *(Solymosi [29]) Let* $\boldsymbol{Z} \in \mathbb{R}^{n \times n}$ *be the set of non-singular matrices. Thus, for many pairs* $\boldsymbol{X}_1, \ \boldsymbol{X}_2 \in \boldsymbol{Z}$*, the sum* $\boldsymbol{Y} = \boldsymbol{X}_1 + \boldsymbol{X}_2$ *is non-singular. i.e.* $\sigma_{min}(\boldsymbol{Y}) \neq 0$*.*

Proposition 2. *Assuming that the optimal solution for a DNN is* $\boldsymbol{\theta}$*. A relative change of the hidden representation at layer* l*,* $\Delta \boldsymbol{H}(\boldsymbol{X})^l$*, results into a relative solution change,* $\Delta\theta$*, which can be expressed as follows*

$$\frac{\| \ \Delta \boldsymbol{\theta} \ \|}{\| \ \boldsymbol{\theta} \ \|} \leq \kappa(\boldsymbol{H}(\boldsymbol{X})^l) \frac{\| \ \Delta \boldsymbol{H}(\boldsymbol{X})^l \ \|}{\| \ \boldsymbol{H}(\boldsymbol{X})^l \ \|} \ : 0 \leq l \leq m, \tag{7}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}^l\}_{l=1}^m$, and $\boldsymbol{H}(\boldsymbol{X})^0 = \boldsymbol{X}$ for $l = 0$ is the input to the DNN. We note that changes in the hidden representation $\parallel \Delta \boldsymbol{H}(\boldsymbol{X})^l \parallel$ can result from small intentional variations (or inevitable noise) that are captured in the input data batch, $\boldsymbol{X}$.

Definition 2. *For an m-layer DNN, let the input to layer l be $\boldsymbol{H}(\boldsymbol{X})^{l-1}$, and the weight and error gradient at iteration t be $\boldsymbol{W}^l(t)$ and $\boldsymbol{\Delta}^l(t)$, respectively. The weight update for $\boldsymbol{W}^l$ at iteration $t+1$ denoted $\boldsymbol{W}^l(t+1)$ is*

$$\boldsymbol{W}^l(t+1) = \boldsymbol{W}^l(t) + \eta \boldsymbol{\Delta}^l(t) \boldsymbol{H}(\boldsymbol{X})^{l-1} : 1 \leq l \leq m, \tag{8}$$

where η is the learning rate.

4.1 PlainNet (PlainNet)

Let the input of a linear m-layer PlainNet be $\boldsymbol{X} \in \mathbb{R}^{n \times N}$, and the hidden layers parameterized by $\boldsymbol{\theta}_p = \{\boldsymbol{W}_p^l \in \mathbb{R}^{n \times n}\}_{l=1}^m$. The output of the PlainNet in the last layer m, $\boldsymbol{H}(\boldsymbol{X})_p^m$, can be written as

$$\boldsymbol{H}(\boldsymbol{X})_p^m = \prod_{l=1}^{m} \boldsymbol{W}_p^l \boldsymbol{W}_p^{l-1} \cdots \boldsymbol{W}_p^2 \boldsymbol{W}_p^1 \boldsymbol{X}. \tag{9}$$

Let $\gamma_{p_{min}}^m = \prod_{l=1}^m \sigma_{min}(\boldsymbol{W}_p^l)$ characterize the cummulative outcome of the product of the minimum singular values of different layer weights, $\sigma_{min}(\boldsymbol{W}_p^l)$. Particularly, $\gamma_{p_{min}}^m$ allows the characterization of the singularity or near-singularity of the hidden representation $\boldsymbol{H}(\boldsymbol{X})_p^m$ in a DNN using Eqn. (5). Considering that $m \gg 1$ for very deep PlainNets, Corrollary 2 where $P(\sigma_{min}(\boldsymbol{W}_p^l) < 1) = 1 : 1 \leq l \leq m$ yields $\gamma_{p_{min}}^m \ll 1$. Importantly, for $m \gg 1$, it is observed that $\gamma_{p_{min}}^m$ can become extremely small such that insufficient machine floating point precision results in a rounding-off to zero. i.e. $\gamma_{p_{min}}^m = 0$. As such, the transformation caused by $\prod_{l=1}^m \boldsymbol{W}_p^l$ collapses space, and is thus singular. Subsequently, using Lemma 1, the result of the transformation of $\boldsymbol{X}$ based on $\prod_{l=1}^m \boldsymbol{W}_p^l$ in Eqn. (9) is singular; that is, $\boldsymbol{H}(\boldsymbol{X})_p^m$ is singular, and $\kappa(\boldsymbol{H}(\boldsymbol{X})_p^m) = \infty$.

Remark 1. From Proposition 2, $\parallel \Delta \boldsymbol{\theta}_p \parallel / \parallel \boldsymbol{\theta}_p \parallel$ is unbounded given $\kappa(\boldsymbol{H}(\boldsymbol{X})_p^m) = \infty$ for the PlainNet. Consequently, small changes in the hidden representation, $\Delta \boldsymbol{H}(\boldsymbol{X})^l$, in Proposition 2 are so magnified that optimization is haphazard in the solution space defined by $\boldsymbol{\theta}_p$. The constant and extreme fluctuation of $\boldsymbol{\theta}_p$ means that optimization cannot progress, and convergence to any decent local minima is impossible. Since $\boldsymbol{H}(\boldsymbol{X})_p^m$ is singular, Lemma 1 shows that the term $\eta \boldsymbol{\Delta}^m(t) \boldsymbol{H}(\boldsymbol{X})^{m-1}$ in Definition 2 is singular too so that the weight update, $\boldsymbol{W}^m(t+1)$ *can become badly conditioned*; this observation is confirmed via experiments in Section 5. In constrast, $\gamma_{p_{min}}^m$ does not become extremely small for shallow PlainNets where typically $m < 20$ such that $\kappa(\boldsymbol{H}(\boldsymbol{X})_p^m) < a : a \in \mathbb{R}$. Hence, $\parallel \Delta \boldsymbol{\theta}_p \parallel / \parallel \boldsymbol{\theta}_p \parallel$ in Proposition 2 for the shallow PlainNet is bounded by a reasonable value, and the fluctuation of $\boldsymbol{\theta}_p$ due to $\Delta \boldsymbol{H}(\boldsymbol{X})^l$ is moderate so that shallow PlainNets can be successfully optimized. i.e. model optimization converges.

4.2 Concatened Network of aggregated hidden representations (ConcNeXt)

The theoretical analysis of the hidden representations of the ConcNeXt show that they are never singular. However, we first give the following important lemma and axiom that allow the characterization of the minimum singular value of the outcome of the concatenation of any matrix and the identity matrix.

Lemma 4. *Let a matrix* $\boldsymbol{B} \in \mathbb{R}^{n \times n_2}$ *be the vertical concatenation of any matrix* $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$ *with* $n_1 \geq n_2$ *and the identity* $\mathbf{I} \in \mathbb{R}^{n_2 \times n_2}$ *as in* $\boldsymbol{B} = [\boldsymbol{A} \oplus \mathbf{I}]$*, so that* $n = n_1 + n_2$*. Let the singular values of* $\boldsymbol{B}$ *and* $\boldsymbol{A}$ *be* $\{\sigma(\boldsymbol{B})_i\}_{i=1}^{n_2}$ *and* $\{\sigma(\boldsymbol{A})_i\}_{i=1}^{n_2}$*, respectively. Then, it can be shown that* $\sigma(\boldsymbol{B})_i = \sqrt{(\sigma(\boldsymbol{A})_i)^2 + 1} : 1 \leq i \leq n_2$.

The implication of Lemma 4 is that $\sigma_{min}(B) \geq 1 \because \sigma_{min}(A) \geq 0$.

Axiom 2. *Let a matrix* $\boldsymbol{B} \in \mathbb{R}^{q \times z}$ *be the vertical concatenation of* s *random matrices* $\boldsymbol{A}_k \in \mathbb{R}^{q_k \times z} : 1 \leq k \leq s$ *with* $q = \sum_{k=1}^{s} q_k$*, as in* $\boldsymbol{B} = \Upsilon_{k=1}^{s} \boldsymbol{A}_k = [\boldsymbol{A}_1 \oplus \cdots \oplus \boldsymbol{A}_k \oplus \cdots \oplus \boldsymbol{A}_s]$*. Then, we conclude that the aggregated matrix,* $\boldsymbol{B}$*, is also a random matrix.*

Let the input of a linear m-layer ConcNeXt be $\boldsymbol{X} \in \mathbb{R}^{n \times N}$, and the hidden layers parameterized by $\boldsymbol{W}_g^1 \in \mathbb{R}^{q \times n}$ and $\{\boldsymbol{W}_g^l \in \mathbb{R}^{q \times q}\}_{l=2}^{m}$, so that $\boldsymbol{\theta}_g = \{\boldsymbol{W}_g^l\}_{l=1}^{m}$. Following Eqn. (4), the output of the ConcNeXt in the last layer m, $\boldsymbol{H}(\boldsymbol{X})_g^m$, can be written as

$$\boldsymbol{H}(\boldsymbol{X})_g^m = \prod_{l=1}^{m} (\Upsilon_{k=1}^{s} \boldsymbol{W}_g^{l,k} \oplus \mathbf{I}) \boldsymbol{X}. \tag{10}$$

Using Axiom 2, $\Upsilon_{k=1}^{s} \boldsymbol{W}_g^{l,k}$ in Eqn. (10) can be aggregated in a compact form so that we have $\boldsymbol{W}_a^l = \Upsilon_{k=1}^{s} \boldsymbol{W}_g^{l,k}$. Therefore, Eqn. (10) becomes

$$\boldsymbol{H}(\boldsymbol{X})_g^m = \prod_{l=1}^{m} (\boldsymbol{W}_a^l \oplus \mathbf{I}) \boldsymbol{X}. \tag{11}$$

Let $\gamma_{g_{min}}^m = \prod_{l=1}^{m} \sigma_{min}(\boldsymbol{W}_a^l \oplus \mathbf{I})$. In Eqn. (11), we have $\sigma_{min}(\boldsymbol{W}_a^l \oplus \mathbf{I}) \geq 1 : 1 \leq l \leq m$ from Lemma 4. As such, $\gamma_{g_{min}}^m \geq 1$ so that the overall transformation effect of $\prod_{l=1}^{m} [\boldsymbol{W}_a^l \oplus \mathbf{I}]$ in Eqn. (11) does not collapse space, and thus is non-singular. Since $\prod_{l=1}^{m} (\boldsymbol{W}_a^l \oplus \mathbf{I})$ is non-singular and $\boldsymbol{X}$ is non-singular from Assumption 1, we can conclude using Lemma 2 that $\boldsymbol{H}(\boldsymbol{x})_g^m$ in Eqn. (11) is non-singular too. i.e $\kappa(\boldsymbol{H}(\boldsymbol{x})_g^m) < a : a \in \mathbb{R}$.

Remark 2. Considering $\kappa(\boldsymbol{H}(\boldsymbol{X})_g^m) < a : a \in \mathbb{R}$, $\| \Delta \boldsymbol{\theta}_g \| / \| \boldsymbol{\theta}_g \|$ in Proposition 2 for the ConcNeXt is bounded. Thus, small changes in the hidden representation, $\Delta \boldsymbol{H}(\boldsymbol{X})^m$, in Proposition 2 results in moderate changes in the solution, $\boldsymbol{\theta}_g$. The stability of $\boldsymbol{\theta}_g$ means that optimization can progress successfully, and decent local minima can be reached. Again, considering that $\boldsymbol{H}(\boldsymbol{X})_g^m$ is non-singular, the term $\eta \boldsymbol{\Delta}^m(t) \boldsymbol{H}(\boldsymbol{X})^{m-1}$ in Definition 2 is non-singular by applying Lemma 2. Finally, noting that $\boldsymbol{W}_g^m$ is non-singular from Proposition 1, the weight update $\boldsymbol{W}_g^m(t+1)$ in Eqn. (8) is non-singular with a high probablity using Lemma 3. Our experiments indeed confirm these interesting observations.

5 Experiments

5.1 Experimental settings

This section reports the experimental results that validate the theoretical analysis given in Section 4 using 110 layers PlainNet and ConcNeXt models, which are multilayer perceptrons (MLPs) trained on the MNIST [30] dataset. All model parameters are initialized from Gaussian distributions using the He method [27]. In addition, all models are trained using gradient descent, learning rate = [0.0001-0.1], momentum rate = 0.9, batch normalization (BN), weight decay of 10^{-4}, batch size of 128, and for 60 epochs. Table 1 shows the model architectures and number of parameters for the different models. Note that for showing the agreement between our theoretical analysis and practical models, the experiments reported herein use the rectified linear activation. For additional experiments using the linear activation function, see Section A7.2 in the supplementary material. Furthermore, Section A8 in the supplementary material contains the experiments for convolutional neural network (CNN) models using CIFAR-10 dataset [31]. In the following section, important results for the MLP models are reported.

5.2 Results and discussion

Training results are given in Table 1, where the PlainNet clearly reflects poor optimization. In contrast, ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) are both well optimized. For testing, ConcNeXt-110 ($s = 2$) slightly outperforms ConcNeXt-110 ($s = 1$); similar results for $s = 2$ on the harder CIFAR-10 dataset are provided in Section A8 of the supplementary material. The hidden representations for PlainNet-110 are shown in Fig. 2, where singularity (i.e. units respond in similar fashion for different input data) is seen. In contrast, ConcNeXt-110 ($s = 1$) in Fig. 2 shows no singularity problems (i.e. units respond in different manners for different input data); the hidden representations for ConcNeXt-110 ($s = 2$) is similar to that of ConcNeXt-110 ($s = 1$) so are not shown; Section A7.1 of the supplementary material contains results for ConcNeXt-110 ($s = 2$). Fig. 3 and Fig. 4 show the weight values and hidden units' outputs in the different layers of the MLPs, respectively. It is seen that PlainNet-110 has bizzarely high weight and units' output values, while ConcNeXt-110 ($s = 1$) has reasonable weight and units' output values; ConcNeXt-110 (s=2) has values that are similar to ConcNeXt-110 ($s = 1$), and thus not shown.
Fig. 5 and Fig. 6 show the conditions of the weights and hidden representations in the MLPs, respectively. From Fig. 5 (left), the weights of PlainNet-110 have extremely high conditions numbers, while the weights of ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) both have small condition numbers. From Fig. 5 (right), the weights of PlainNet-110 have zero singular values starting from the 25th layer, while the weights of ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) both have a minimum singular value of one for all the layers. Fig. 6 (left) shows that the hidden representations of PlainNet-110 have extremely high and even infinite condition numbers that plague optimization, while ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) both have small condition numbers that foster successful optimization. For clarity, Fig. 6 (right) shows that the hidden representations of ConcNeXt-110 ($s = 2$) operates with smaller

MLP model	PlainNet-110	ConcNeXt-110 ($s = 1$)	ConcNeXt-110 ($s = 2$)
Hidden units per layer	180	50	25
Parameters	3.66M	3.92M	3.77M
Training accuracy	13.22%	100%	100%
Testing accuracy	12.45%	98.42%	98.83%

Table 1: 110 layers MLP details and results on the MNIST dataset.

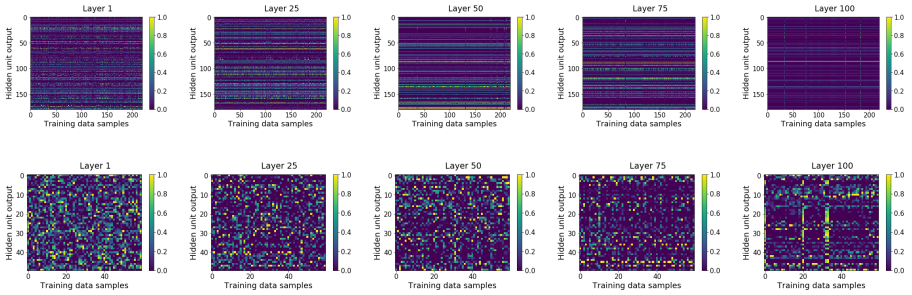

Fig. 2: Hidden representations for the 110 layers MLPs using randomly chosen batch of input data from the MNIST dataset. Top row: PlainNet-110. Bottom row: ConcNeXt-110 ($s = 1$).

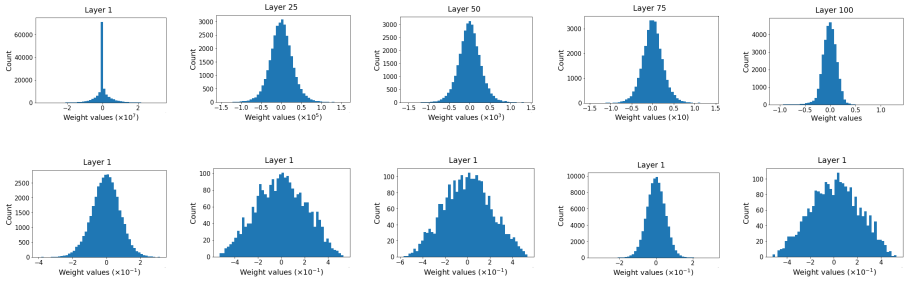

Fig. 3: Weights distribution for the 110 layers MLPs using randomly chosen batch of input data from the MNIST dataset. Top row: PlainNet-110. Bottom row: ConcNeXt-110 ($s = 1$).

condition numbers as compared to the ConcNeXt-110 ($s = 1$). Subsequently, using this observation in Proposition 2 explains the improved generalization performance of ConcNeXt-110 ($s = 2$) over ConcNeXt-110 ($s = 1$).

5.3 Main insights into models concatenated hidden representations

The main insights from the theoretical and empirical results are summarized as follows.

1. The minimum singular values for the weights in the PlainNet are invariably less than one. i.e. $\sigma_{min}(\boldsymbol{W}_p^l) < 1 : 1 \leq l \leq m$. As such, for very DNNs where $m \gg 1$, the repeated multiplication of the input data, $\boldsymbol{X}$, by the model weights $\{\boldsymbol{W}_p^1, \cdots, \boldsymbol{W}_p^m\}$ causes some components of $\boldsymbol{X}$ to vanish so that collinearity and thus singularity occur in the resulting output $\boldsymbol{H}(\boldsymbol{X})_p^m$. Finally, singularity plagues model optimization.

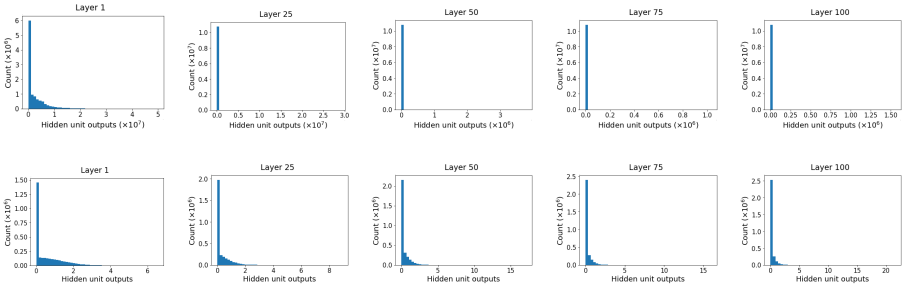

Fig. 4: Units' outputs distribution for the 110 layers MLPs using randomly chosen batch of input data from the MNIST dataset. Top row: PlainNet-110. Bottom row: ConcNeXt-110 ($s = 1$).

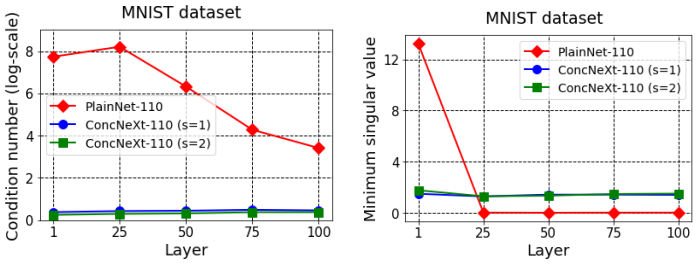

Fig. 5: Condition of the layer weight in the MLPs. Left: condition number of the weights in the different layers *plotted to log-scale*. Right: smallest singular value of the weights in the different layers.

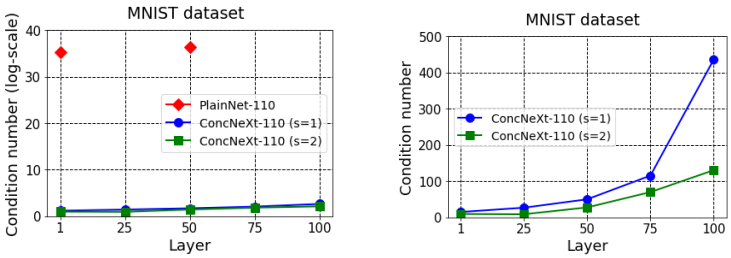

Fig. 6: Condition number of the hidden layer representations in the 110 layers MLPs. The layers for which values are not shown have infinite condition numbers. Left: condition number of the hidden representations in the different layers. Right: condition number of the hidden representations in the different layers of the ConcNeXt ($s = 1$) and ConcNeXt ($s = 2$).

2. Skip connections that concatenate hidden representations operate such that the minimum singular values for the compactly expressed model weights are $\sigma_{min}(\boldsymbol{W}_a^l \oplus \mathbf{I}) \geq 1 : 1 \leq l \leq m$. Thus, for very DNNs where $m \gg 1$, the repeated multiplication of the input data, $\boldsymbol{X}$, by the model weights $\{(\boldsymbol{W}_a^1 \oplus \mathbf{I}), \cdots, (\boldsymbol{W}_a^m \oplus \mathbf{I})\}$ does not cause any component of $\boldsymbol{X}$ to vanish, and thus collinearity and singularity are mitigated. Consequently, such models with several layers can be successfully optimized.

6 Conclusion

The successful training of very DNNs requires using skip connections of various forms. However, understanding why DNN architectures with skip connections circumnavigate optimization problems is challenging. Specifically, the operation of DNNs that employ the concatenation of hidden representations is so complicated that their theoretical analysis is outrightly missing in the literature. This paper presents the theoretical analysis of DNNs with skip connections and concatenated hidden representations, which to the best of our knowledge is the first in the literature. Our theoretical results that are confirmed via extensive experiments show that concatenating hidden representations improve the condition of the hidden representations by mitigating singularity problems in hidden representations that ensue from the repeated multiplication of model weights.

Acknowledgments

This work was funded by the National Research Fund (FNR), Luxembourg, under the project reference R-AGR-0424-05-D/Björn Ottersten and CPPP17/IS/11643091/IDform/Aouada.

References

1. C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*, 2015, pp. 562–570.
2. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.
3. R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Conference on Learning Theory*, 2016, pp. 907–940.
4. I. Safran and O. Shamir, "Depth-width tradeoffs in approximating natural functions with neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2979–2987.
5. O. K. Oyedotun, A. El Rahman Shabayek, D. Aouada, and B. Ottersten, "Highway network block with gates constraints for training very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1658–1667.
6. R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.
7. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
8. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
9. O. K. Oyedotun, D. Aouada, B. Ottersten *et al.*, "Training very deep networks via residual learning with stochastic input shortcut connections," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 23–33.
10. A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in neural information processing systems*, 2016, pp. 550–558.

11. D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" in *International Conference on Machine Learning*, 2017, pp. 342–350.
12. K. Greff, R. K. Srivastava, and J. Schmidhuber, "Highway and residual networks learn unrolled iterative estimation," in *International Conference Learning Representations*, 2017.
13. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
14. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
15. G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," in *International Conference on Learning Representations*, 2017.
16. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
17. S. Jastrzkebski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio, "Residual connections encourage iterative inference," in *International Conference on Learning Representations*, 2018.
18. B. Chang, L. Meng, E. Haber, F. Tung, and D. Begert, "Multi-level residual networks from dynamical systems view," in *International Conference on Learning Representations*, 2018.
19. Y. Zhou and Y. Liang, "Critical points of neural networks: Analytical forms and landscape properties," in *International Conference on Learning Representations*, 2017.
20. S. Sonoda and N. Murata, "Transport analysis of infinitely deep neural network," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 31–82, 2019.
21. Q. Nguyen and M. Hein, "Optimization landscape and expressivity of deep cnns," in *International Conference on Machine Learning*, 2018, pp. 3730–3739.
22. T. Laurent and J. Brecht, "Deep linear networks with arbitrary loss: All local minima are global," in *International Conference on Machine Learning*, 2018, pp. 2902–2907.
23. K. Kawaguchi, "Deep learning without poor local minima," in *Advances in neural information processing systems*, 2016, pp. 586–594.
24. A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *International Conference on Learning Representations*, 2014.
25. A. Neubauer, "A new gradient method for ill-posed problems," *Numerical Functional Analysis and Optimization*, vol. 39, no. 6, pp. 737–762, 2018.
26. A. Neubauer and O. Scherzer, "A convergence rate result for a steepest descent method and a minimal error method for the solution of nonlinear ill-posed problems," *Zeitschrift für Analysis und ihre Anwendungen*, vol. 14, no. 2, pp. 369–377, 1995.
27. K. He, X. Zhang, S. Ren, and 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.
28. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.
29. J. Solymosi, "The sum of nonsingular matrices is often nonsingular," *Linear Algebra and its Applications*, vol. 552, pp. 159–165, 2018.
30. Y. LeCun and C. Cortes, "Mnist handwritten digit database," http://yann.lecun.com/exdb/mnist/, Last accessed, October. 2019.
31. A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10, cifar-100 (canadian institute for advanced research)," http://www.cs.toronto.edu/~kriz/cifar.html, Last accessed, October. 2019.

Why do Deep Neural Networks with Skip Connections and Concatenated Hidden Representations Work?

Oyebade K. Oyedotun ✉ and Djamila Aouada

Interdisciplinary Centre for Security, Reliability and Trust (SnT),
University of Luxembourg, L-1855 Luxembourg
{oyebade.oyedotun,djamila.aouada}@uni.lu

http://wwwen.uni.lu/snt

A1 Proof of Proposition 1

Let $W = \{\boldsymbol{w}_i\}_{i=1}^{n}$, where $\boldsymbol{w}_i$ is the i-th vector in W, and its entries are randomly sampled from a continuous uniform or Gaussian distribution. Given $\boldsymbol{w}_i$, the matrix W excluding $\boldsymbol{w}_i$ is denoted W_{-i}; thus, the span of W_i is $W_{-i}^{span} = span(\boldsymbol{w}_1, .., \boldsymbol{w}_{n-1})$. As such, validating *Proposition 1* means showing that any given $\boldsymbol{w}_i$ does not lie in the span of W_{-i}^{span}. i.e. $\boldsymbol{w}_i$ and W_{-i}^{span} are linearly independent.
To start with, it is trivial to observe that $P(\boldsymbol{w}_1 \neq 0) = 1$, as the Lebesgue measure of any singleton set is zero [1]; hence, $P(w_1 \notin W_{-i}^{span}) = 1$. In addition, for $p \in \{2, ..., n-1\}$, let $W_{-1,p} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_p\}$ be the set of first p vectors, excluding vector i, such that $W_{-1,p}$ is linearly independent with a probability of 1. Thus, we can conclude with a probablity of 1 that $W_{-1,p}$ spans the p-dimensional subspace of $\mathbb{R}^n$, and therefore also has a Lebesgue measure of zero. Importantly, $\boldsymbol{w_{p+1}}$ resides on $\mathbb{R}^n$, and hence is on the exterior of the subspace with a probability of 1. □

A2 Proof of Corollary 2

We rely on extensive empirical observations for the proof of *Corollary 2*. The objective is to show that given the fashion in which the DNN weight at an arbitrary layer l, W^l, is initialized, the smallest singular value is less than one; that is, $\exists\ \sigma_{min}(W^l) < 1$ with a probability of one. In fact, we show that there are often several singular values that are less than one. In order to achieve this, several experiments are performed using popular initialization methods such the He method in [2] and random initializations. For exhaustiveness, weight values drawn from uniform and Gaussian distributions using both methods are considered. Furthermore, matrices of different sizes are considered for the experiments. Lastly, to improve results reliability, each experiment is repeated 1000 times. The different initialization schemes employed, along with their implementation details are summarized as follows.

1. **Gaussion distribution using He method:** The weight values $w_{ij}^l \in W^l$ are drawn from a Gaussian distribution using the He method [2] as follows

$$w_{ij}^l \sim \mathcal{N}(\mu = 0,\ std = \sqrt{2/n_{in}^l}), \tag{1}$$

where n_{in}^l is the number of units feeding into layer l. See Fig. A1 for results.

2. **Uniform distribution using He method:** The weight values $w_{ij}^l \in W^l$ are drawn from a uniform distribution using the He method [2] as follows

$$w_{ij}^l \sim U[\sqrt{6/n_{in}^l}, -\sqrt{6/n_{in}^l}], \tag{2}$$

where n_{in}^l is the number of units feeding into layer l. See Fig. A2 for results.

3. **Uniform distribution:**:The weight values $w_{ij}^l \in W^l$ are drawn from a uniform distribution in the range $-r$ to r as follows

$$w_{ij}^l \sim U[-r, r]. \tag{3}$$

The value $r = 0.05$ is used for the reported experiments; see Fig. A3. However, it is observed that similar results are obtained using other reasonable values for r.

4. **Gaussion distribution:** The weight values $w_{ij}^l \in W^l$ are drawn from a Gaussian distribution as follows

$$w_{ij}^l \sim \mathcal{N}(\mu = 0,\ std = \beta), \tag{4}$$

The reported experiments use standard deviation values, $\beta = 0.01$ and $\beta = 0.05$ with $\mu = 0$ that are typical for neural networks; see Fig. A4 and Fig. A5. Again, it is noted that similar results are obtained using other reasonable values for β.

In proofing Corollary 2, Fig. A1 to Fig. A5 show that irrespective of initialization method or matrix dimension, there are several singular values with values, which are less than one. It is crucial to note that for all the experiments, none of the singular values is zero. This observation further strengthens the claim in *Proposition 1* about the non-singularity of random matrices. Furthermore, we report the median of the singular values, which are less than one for each experiment.

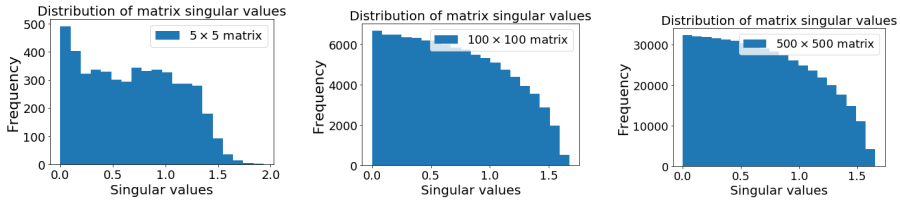

Fig. A1: The distribution of singular values of random matrices with entries sampled from uniform distribution obtained using the He method [2]; each experiment is repeated 1000 times. The median of the singular values that are less than one are as follows. Left: 0.4711. Middle: 0.4728. Right: 0.4734.

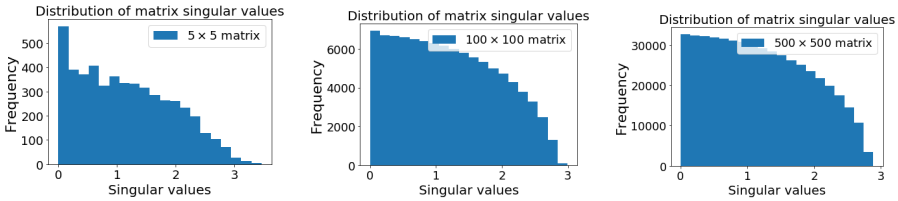

Fig. A2: The distribution of singular values of random matrices with entries sampled from Gaussian distribution obtained using the He method [2]; each experiment is repeated 1000 times. The median of the singular values that are less than one are as follows. Left: 0.4450. Middle: 0.4887. Right: 0.4914.

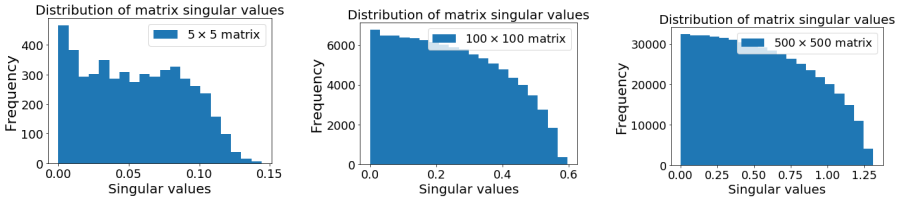

Fig. A3: The distribution of singular values of random matrices with entries sampled from uniform distribution in the range -0.05 to 0.05; each experiment is repeated 1000 times. The median of the singular values that are less than one are as follows. Left: 0.0531. Middle: 0.2336. Right: 0.4537.

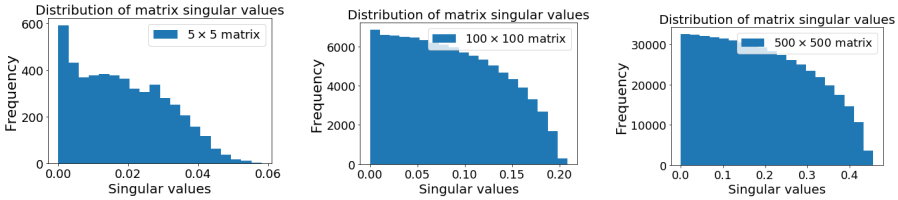

Fig. A4: The distribution of singular values of random matrices with entries sampled from Gaussian distribution with $\mu = 0$ and $\beta = 0.01$; each experiment is repeated 1000 times. The median of the singular values that are less than one are as follows. Left: 0.0173. Middle: 0.0806. Right: 0.1806.

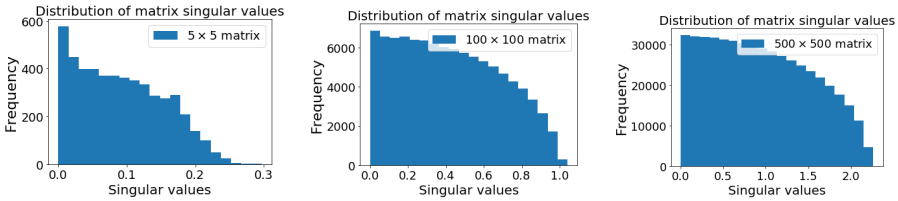

Fig. A5: The distribution of singular values of random matrices with entries sampled from Gaussian distribution with $\mu = 0$ and $\beta = 0.05$; each experiment is repeated 1000 times. The median of the singular values that are less than one are as follows. Left: 0.0856. Middle: 0.4027. Right: 0.4863.

A3 Proof of Lemma 1

Some row(s) or column(s) of V are linearly correlated. As such, $\exists$ a vector $u \in \mathbb{R}^n$ with $\| u \|_2 \neq 0$: $Vu = 0$. Hence, $XVu = 0$, and thus XV is singular. $\square$

A4 Proof of Lemma 2

All row(s) or column(s) of both X and V are linearly uncorrelated. As such, $\nexists$ a vector $u \in \mathbb{R}^n$ with $\| u \|_2 \neq 0$ such that $Xu = 0$ or $Vu = 0$. Consequently, $XVu \neq 0$, and thus XV is non-singular. $\square$

A5 Proof of Proposition 2

First, let $\theta \in \mathbb{R}^{c \times n}$, $X \in \mathbb{R}^{n \times r}$ and $Y \in \mathbb{R}^{c \times r}$. Now, let us consider the simple problem, $Y = \theta X$: $X^{\dagger}$ is the pseudoinvrese of X. The objective is to estimate the solution, θ, given X and Y. Furthermore, let a *small* perturbation of ΔX result in a *small* solution perturbation, $\Delta\theta$, so that

$$Y = (\theta + \Delta\theta)(X + \Delta X). \tag{5}$$

Observing that $\Delta\theta\Delta X \approx 0$ and $Y = \theta X$, (5) becomes

$$\frac{\Delta\theta}{\theta} = -X^{\dagger}\Delta X. \tag{6}$$

Employing Cauchy-Schwarz inequality in (6) yields

$$\frac{\| \Delta\theta \|}{\| \theta \|} \leq \| X^{\dagger} \| \| \Delta X \| \leq \| X^{\dagger} \| \| \Delta X \| \frac{\| X \|}{\| X \|}. \tag{7}$$

In conclusion, taking $\kappa(X) \approx \| X^{\dagger} \| \| X \|$, we obtain

$$\frac{\| \Delta\theta \|}{\| \theta \|} \leq \kappa(X) \frac{\| \Delta X \|}{\| X \|}. \square \tag{8}$$

A6 Proof of Lemma 4

Let matrix $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$ with $n_1 \geq n_2$ have the singular values $\{\sigma(\boldsymbol{A})_1, \cdots, \sigma(\boldsymbol{A})_i, \cdots, \sigma(\boldsymbol{A})_{n_2}\}$, so that the positive semidefinite matrix $\boldsymbol{A}^T\boldsymbol{A}$ have the eigenvalues $\{\lambda(\boldsymbol{A})_1, \cdots, \lambda(\boldsymbol{A})_i, \cdots, \lambda(\boldsymbol{A})_{n_2}\}$. Subsequently, $\boldsymbol{A}$ has singular values $\sigma(\boldsymbol{A})_i = \sqrt{\lambda(\boldsymbol{A}^T\boldsymbol{A})_i}$. Furthermore, given that $\boldsymbol{B} = [\boldsymbol{A} \oplus \mathbf{I}] = \begin{bmatrix} \boldsymbol{A} \\ \mathbf{I} \end{bmatrix}$, we have $\boldsymbol{B}^T\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}^T & \mathbf{I}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{A} \\ \mathbf{I} \end{bmatrix} = \boldsymbol{A}^T\boldsymbol{A} + \mathbf{I}$. Noting that $\lambda(\boldsymbol{A}^T\boldsymbol{A})_i = (\sigma(\boldsymbol{A})_i)^2$, the singular values for $\boldsymbol{B}$, $\sigma(\boldsymbol{B})_i = \sqrt{(\sigma(\boldsymbol{A})_i)^2 + 1}$.
We perform experiments to empirically validate lemma 4, and show the practical implication for the singular values of initialized weight matrices. Specifically, we use matrices of different sizes with values drawn from uniform and Gaussian distribution using the He initialization method [2]. See Fig. A6 and Fig. A7, which show that the smallest singular value for any matrix is one. It is shown in Section A7 that this nice property for initialized weights can be related to the success of the ConcNeXt.

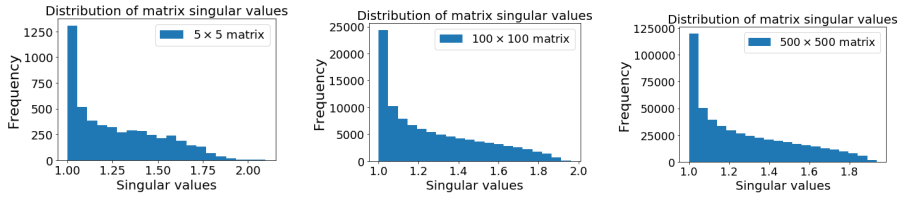

Fig. A6: The distribution of singular values of random matrices with entries sampled from uniform distribution obtained using He method [2], and concatenated with an identity matrix of suitable size; each experiment is repeated 1000 times. The smallest singular value for any matrix is 1.

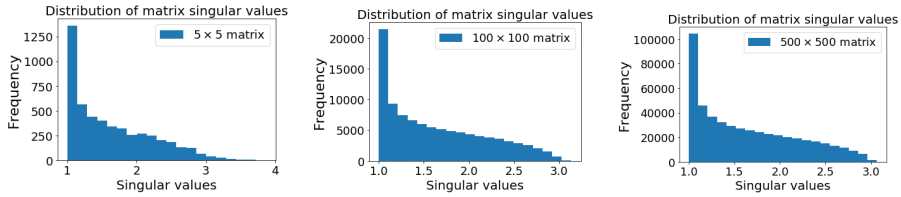

Fig. A7: The distribution of singular values of random matrices with entries sampled from Gaussian distribution obtained using He method [2], and concatenated with an identity matrix of suitable size; each experiment is repeated 1000 times. The smallest singular value for any matrix is 1.

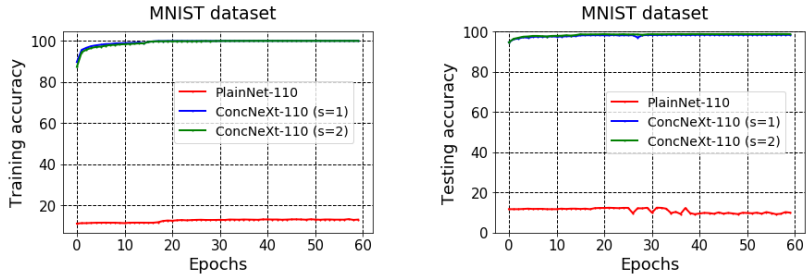

Fig. A8: Training and testing curves for the MLPs with 110 layers using MNIST dataset.

A7 Additional Experiments using Multilayer Perceptron (MLP)

A7.1 Additional Results for models with the rectified activation function

The training and testing curves for the different MLPs reported in Table 1 in the main paper are given in Fig. A8. It is seen that PlainNet-110 training and testing accuracies never exceeded 15% during optimization. However, ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) both show no optimization problem right from the start of training. From Fig. A8 (right), it is observed that the ConcNeXt-110 ($s = 2$) slightly generalizing better than ConcNeXt-110 ($s = 1$); see Table 1 in the main paper.
Given a randomly choosen batch of input data, the outputs of the units in the different

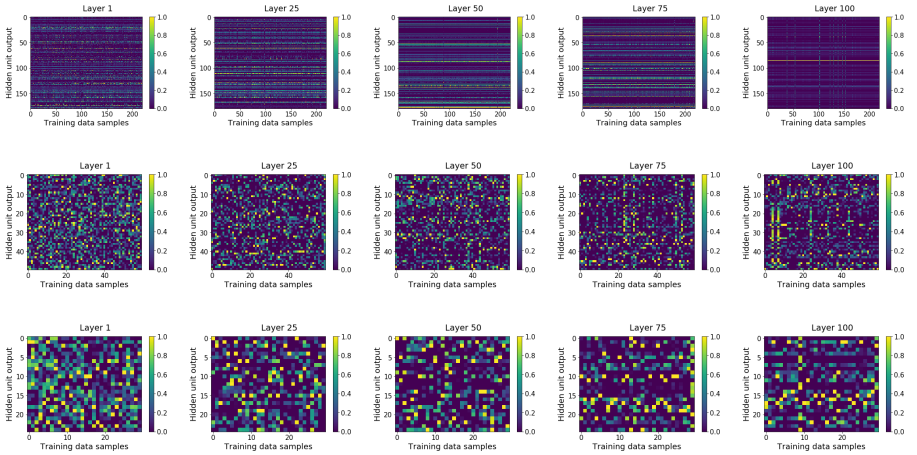

Fig. A9: Hidden layer representations for the 110 layers MLPs trained on MNIST dataset; results use a randomly chosen batch of input data. Top row: PlainNet results. Middle row: ConcNeXt-110 ($s = 1$) results. Bottom row: ConcNeXt-110 ($s = 2$) results.

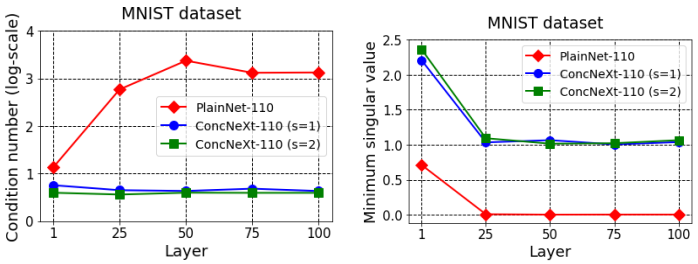

Fig. A10: Condition of the layer weights in the 110 layers MLPs with the *linear activation function*. Left: condition number of the weights in the different layers *plotted to log-scale*. Right: smallest singular value of the weights in the different layers.

hidden layers of the 110 layers MLP PlainNet, ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) are shown in Fig. A9. These experimental results are additional evidence that support the singularity (i.e. collinearity of the hidden representations) in the PlainNet, and non-singularity (i.e. no collinearity) of the hidden representations in the ConcNeXt models with $s \geq 1$. Ultimately, these observations corroborate *Remark 1* and *Remark 2* in the main paper.

A7.2 Results for models with the linear activation function

Herein, the experimental results are obtained using the linear activation function for the different MLP models to show that similar results are obtained in comparision with the models that employed the rectified linear activation function. The same model architectures and training settings as in the models with ReLUs are used. Fig. A10 shows the

CNN model	PlainNet-110	ConcNeXt-110 ($s = 1$)	ConcNeXt-110 ($s = 2$)
Parameters	1.74M	1.73M	1.71M
Training accuracy	10.26%	99.94%	99.95%
Testing accuracy	10.05%	94.51%	94.86%

Table A1: Model details and results of the CNN models with 110 layers trained on CIFAR-10 dataset.

condition of the weights in the different hidden layers for the MLP models. Similar to the models with ReLUs, the condition numbers of the PlainNet weights are significantly higher than those for the ConcNeXt models; see Fig. A10 (left). In addition, from Fig. A10 (right), the minimum singular values for the layer weights in the PlainNet is zero starting from the 25th layer. However, the ConcNeXt models have a minimum singular value of one for the layers. Importantly, these observations validate the bad conditioning of layer weights in very deep PlainNets. In contrast, the ConcNeXt circumnavigate this problem by operating with considerably smaller condition numbers.

A8 Experiments using Convolutional neural Network (CNN)

In order to further experimentally validate the analytical results, convolutional neural Network (CNN) with 110 layers are constructed for the PlainNet and ConcNeXt. The PlainNet is based on the ResNeXt architecture [3], where the skip connections are eliminated. The ConcNeXt is also based on the ResNeXt architecture, where the summation operation is simply replaced with the concatenation operation. For these models, experiments are performed using the harder CIFAR-10 dataset. This allows us to again show that the PlainNet is untrainable, but the ConcNeXt models can be successfully trained. Furthermore, we again observe the improved generalization of the ConcNeXt ($s = 2$) over ConcNeXt ($s = 1$). All the CNN experiments employ rectified linear units (ReLUs), batch normalization (BN), weight decay of 1×10^{-4} and 300 epochs. All layer weights are initialized from Gaussian distributions using the He initialization technique [2].

We note that the hidden representations (i.e. layer outputs) and weights in CNNs are tensors. As such, to obtain the singular values of the hidden representations and weights, we use Higher Order SVD (HOSVD) [4] that generalizes the conventional SVD for matrices to tensors. Finally, the condition numbers for the hidden representations and weights of the models are computed using the ratio of the maximum singular value to the minimum singular value.

Table A1 shows the training results for the different CNN models trained on the CIFAR-10 dataset. It is observed that the PlainNet has poor training and testing accuracies; this reflects poor optimization. In contrast, ConcNeXt-110 ($s = 1$) and ConcNeXt-110 ($s = 2$) models both have good training and testing accuracies; this shows successful optimization. Particularly, we note that ConcNeXt-110 ($s = 2$) with a test accuracy of 94.86% again outperforms ConcNeXt-110 ($s = 1$) with a test accuracy of 94.51%. This observation is similar to that seen in the ResNeXt [3], where cardinality increase resulted in improved model generalization.

Fig. A11 (left) shows the condition numbers of the layer weights in the different CNN

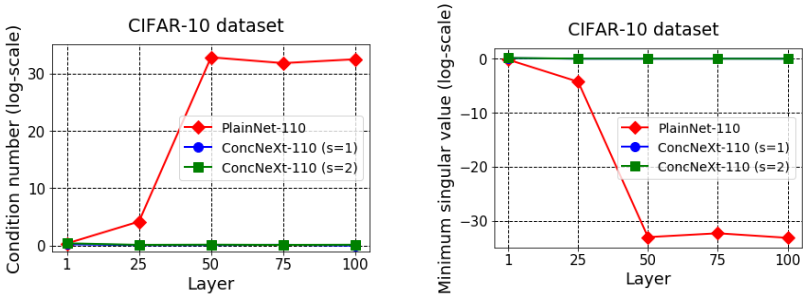

Fig. A11: Condition of the layer weights in the 110 layers CNN models. Left: condition number of the weights in the different layers. Right: smallest singular value of the weights in the different layers.

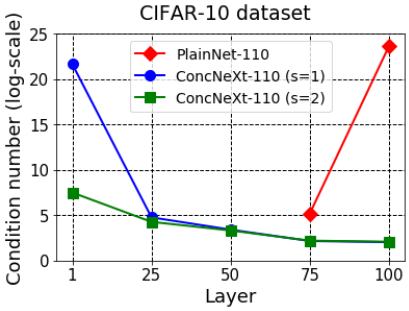

Fig. A12: Condition number of the hidden layer representations in the 110 layers CNN models. The layers for which values are not shown have infinite condition numbers.

models, where it is seen that the PlainNet operates with very high condition numbers. In contrast, the ConcNeXt models operate with small condition numbers. Importantly, Fig. A11 (right) shows the minimum singular values for the layer weights in the models; note that the figure is drawn to log-scale. Therefore, it is seen that the PlainNet weights have roughly zero minimum singular values starting from the 50th layer (i.e. $\log 10^{-33} \approx -33$). However, the layer weights in ConcNeXt models have minimum singular values that are roughly one (i.e. $\log 1 = 0$). In Fig. A12, the condition numbers for the hidden representations of the different layers in the CNN models are shown. For the PlainNet, it is seen that the hidden representations for the 1st, 25th and 50th layers have infinite condition numbers, which is the worst scenario for optimization problems; the 75th has a moderate condition number, while the 100th layer has a very high condition number. In contrast, all the hidden representations in the different layers of the ConcNeXt models (i.e. for $s = 1$ and $s = 2$) have finite condition numbers. Furthermore, it is observed that the hidden representations in the ConcNeXt-110 ($s = 2$) have smaller condition numbers than ConcNeXt-110 ($s = 1$); this can again be related to the improved test accuracy of ConcNeXt-110 ($s = 2$) as compared to ConcNeXt-110 ($s = 1$); see Table A1.

References

1. J. M. Briggs and T. Schaffter, "Measure and cardinality," *The American Mathematical Monthly*, vol. 86, no. 10, pp. 852–855, 1979.
2. K. He, X. Zhang, S. Ren, and 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.
3. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
4. G. Bergqvist and E. G. Larsson, "The higher-order singular value decomposition: Theory and an application [lecture notes]," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 151–154, 2010.