Towards an Optimal Control Perspective of ResNet Training

Jens Püttschneider JENS.PUETTSCHNEIDER@TU-DORTMUND.DE
Institute of Energy Systems, Energy Efficiency and Energy Economics, TU Dortmund, Germany

Simon Heilig SIMON.HEILIG@RUB.DE
Department of Computer Science, Ruhr University Bochum, Germany

Asja Fischer ASJA.FISCHER@RUB.DE
Department of Computer Science, Ruhr University Bochum, Germany

Timm Faulwasser TIMM.FAULWASSER@IEEE.ORG
Institute of Control Systems, Hamburg University of Technology, Germany

Abstract

We propose a training formulation for ResNets reflecting an optimal control problem that is applicable for standard architectures and general loss functions. We suggest bridging both worlds via penalizing intermediate outputs of hidden states corresponding to stage cost terms in optimal control. For standard ResNets, we obtain intermediate outputs by propagating the state through the subsequent skip connections and the output layer. We demonstrate that our training dynamic biases the weights of the unnecessary deeper residual layers to vanish. This indicates the potential for a theory-grounded layer pruning strategy.

1. Introduction

Analyzing properties of neural networks and their training dynamics is one of the promising avenues towards trustworthy applications of modern AI. In this work, we take the optimal control perspective of neural network training in which the forward propagation of the data over the layers of the network is considered as the state trajectory of a dynamical system [2–4, 12]. The trainable parameters act as control signals that optimally steer the state towards a target representation. This way, the training emulates the numerical solution to an optimal control problem, resulting in parameters that minimize the loss function evaluated for the terminal state. The systems and control perspective is particularly insightful for ResNets [5], since they represent Euler forward discretizations of continuous-time neural ODEs [1]. Recent works on optimal control training of ResNets suggest augmenting the training objective by the loss function evaluated for intermediate outputs placed at the states of the hidden residual blocks [3, 11, 12]. From an optimal control perspective, this corresponds to a stage cost in the training objective [1]. Placed within the interpolation regime, the training formulation with stage cost gives rise to a self-regularizing ResNet possessing asymptotic behaviors, in which, with sufficient depth, the weights decay to zero and the hidden states approach their target state [2–4, 12]. However, the dynamic systems perspective on ResNets only addresses architectures with a fixed latent dimension. While the work [3] investigates varying latent dimensions via a continuous

1. The stage cost loss from optimal control is similar to early exiting, where intermediate predictions are obtained from hidden states. In joint training formulations, similar to optimal control, the loss functions corresponding to the intermediate predictions are included in the training objective [9, 10, 13].

© J. Püttschneider, S. Heilig, A. Fischer & T. Faulwasser.

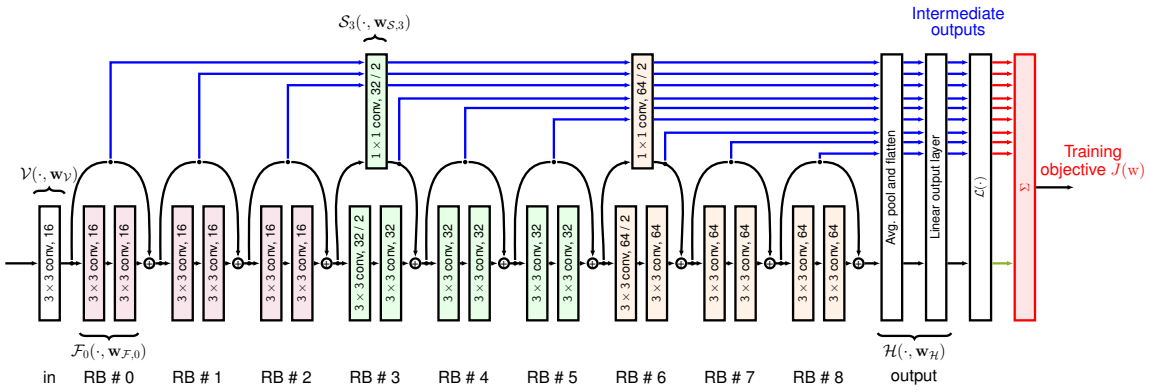


Figure 1: Illustration of a ResNet-v2 architecture while training with our proposed stage cost regularizer penalizing intermediate outputs. We propose to forward the hidden states through the network's non-trivial skip connections to account for dimensionality projections.

space-time neural network, an optimal control perspective to real-world architectures remains an open problem. Moreover, existing results on stage cost formulations for ResNet training place somewhat restrictive assumptions on the loss function, thereby excluding the most commonly used one – the cross-entropy with binary targets.

In this paper, we show how this optimal control formulation with stage cost can be extended to more general architectures, including standard ResNets [5] and generic loss functions. For ResNet architectures with 1x1 convolutions in the skip connections, incorporating skip connections in the intermediate outputs and including their respective losses as a stage cost recovers a training scheme following the solution of an optimal control problem. By introducing the notion of shallower *SubResNets*, we prove that the performance of the intermediate outputs of the stage cost ResNet is asymptotically bounded by the performance of the shallower SubResNet trained using standard training. Our asymptotic results show that residual blocks tend to learn identity mappings after forward propagation sufficiently accomplishes the learning task, hence self-regularizing the depth of the network. Empirically, we find that our ResNet trained with stage cost converges much faster in terms of depth to near near optimal values of the loss function. In short, our paper makes the following contributions:

- We propose a training scheme for residual-connection based neural networks derived from an optimal control perspective, including stage cost evaluation of loss functions.
- We derive asymptotic results on the loss dynamics, pointing towards self-regularizing training in terms of optimal depth.
- We empirically assess the convergence of the loss and quantify our loss bounds on standard benchmarks.

2. ResNet Notation

We briefly revisit the core components of the ResNet-v2 [5, 6] architecture which is the canvas for this work.

As illustrated in Figure 1, the forward pass goes as follows: a sample $x^{(i)}$ is initially embedded into a latent space via $\mathcal{V}(x^{(i)}, \mathbf{w}_{\mathcal{V}})$, then N residual blocks process the input to eventually forward the final representation $x_N^{(i)}$ to the output head $\mathcal{H}(x_N^{(i)}, \mathbf{w}_{\mathcal{H}})$. We denote trainable weights

as $\mathbf{w} = [\mathbf{w}_{\mathcal{V}}, \mathbf{w}_0, ..., \mathbf{w}_{N-1}, \mathbf{w}_{\mathcal{H}}]$, where $\mathbf{w}_k = [\mathbf{w}_{\mathcal{F},k}, \mathbf{w}_{\mathcal{S},k}]$ corresponds to the kth residual block $f_k(x_k^{(i)}, \mathbf{w}_k)$ that is composed of the convolutional layers $\mathcal{F}_k(x_k^{(i)}, \mathbf{w}_{F,k})$ and the skip connection $\mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k})$ with optional parameters $\mathbf{w}_{\mathcal{S},k}$. If two consecutive blocks have the same dimensionality, an identity skip connection is employed, but if there is a change in dimensionality, a 1x1 convolution is used as a projection layer. For the ease of reading, we denote $\mathcal{K}_{\mathcal{S}}$ as the index set of blocks having non-trivial skip connections. In summary, a ResNet-N with N residual blocks comprises the following system:

$$\begin{aligned} x_0^{(i)} &= \mathcal{V}(x^{(i)}, \mathbf{w}_{\mathcal{V}}) \ , \\ x_{k+1}^{(i)} &= f_k(x_k^{(i)}, \mathbf{w}_k) = \begin{cases} x_k^{(i)} & +\mathcal{F}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{F},k}), \ k \notin \mathcal{K}_{\mathcal{S}} \\ \mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}) & +\mathcal{F}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{F},k}), \ k \in \mathcal{K}_{\mathcal{S}}, \end{cases} \quad \forall k = 0, ..., N-1 \ , \\ \hat{y}^{(i)} &= \mathcal{H}(x_N^{(i)}, \mathbf{w}_{\mathcal{H}}) \ . \end{aligned} \tag{1}$$

A batch of D samples $\mathbf{x} = \left[x^{(1)}, x^{(2)}, \ldots, x^{(D)}\right]$ and its propagation through the network is denoted by boldface variables. Assuming a constant state dimension in the hidden residual blocks, the propagation of the data in the residual blocks can be understood as a time-varying dynamical system: $\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{x}_k + \mathcal{F}_k(\mathbf{x}_k, \mathbf{w}_{\mathcal{F},k})$, with stacked data $\mathbf{x}_k$ as state variable and the trainable parameters $\mathbf{w}_k$ as control inputs. The objective function of standard neural network training is the loss $\mathcal{L}$ at the network output, which corresponds to the terminal penalty in optimal control. In addition to common weight decay, optimal control approaches to ResNet training have suggested including a stage cost regularizer $\ell(\mathbf{x}_k, \mathbf{w}_k)$ that depends also on the states $\mathbf{x}_k$ [3, 4, 11, 12]. This leads to the training objective $J(\mathbf{w}) = \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, \mathbf{w}_k) + \mathcal{L}(\hat{\mathbf{y}})$.

3. Optimal Control Perspective on Standard ResNets

We now extend the optimal control formulation of ResNet training to standard ResNets with non-identity skip connections [5] that allows for a theoretical analysis of the training. To invoke the training objective using standard ResNet (1), we need a tractable notion of the stage costs $\ell(\mathbf{x}_k, \mathbf{w}_k)$. To this end, we design the intermediate outputs to reuse the existing loss function based on the following observation.

Consider a ResNet architecture (1) with N residual blocks, in which the parameters of the residual branches for $M \leq k \leq N-1$ are set to zero $\mathbf{w}_{\mathcal{F},k} = 0$. Then, the hidden state $x_M^{(i)}$ is forwarded only by the skip connections of the following residual blocks, resulting in the output $\hat{y}^{(i)} = \mathcal{H}(\mathcal{S}_{N-1}(\cdots \mathcal{S}_M(x_M^{(i)}, \mathbf{w}_{\mathcal{S},M}), \cdots, \mathbf{w}_{\mathcal{S},N-1}), \mathbf{w}_{\mathcal{H}})$. Moreover, since most residual blocks have identity skip connections, the remaining number of layers the state $x_M^{(i)}$ gets propagated is comparatively low. Hence, we can use those non-trivial skip connections as building blocks for obtaining the intermediate output of state $x_M^{(i)}$, since, similar to the standard design of early exits, they should consist of few layers for computational efficiency [10, 13]. This leads us to the definition of intermediate output heads $\mathcal{E}_k$, which predict $\hat{y}_k^{(i)}$ based on intermediate states $x_k^{(i)}$ via:

$$\hat{y}_k^{(i)} = \mathcal{E}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{E},k}) = \mathcal{H}(\mathcal{S}_{N-1}(\cdots \mathcal{S}_k(x_k^{(i)}, \mathbf{w}_{\mathcal{S},k}), \cdots, \mathbf{w}_{\mathcal{S},N-1}), \mathbf{w}_{\mathcal{H}}) \quad \forall k = 0, \ldots, N-1. \tag{2}$$

Figure 1 illustrates the proposed flow of data through the skip connections of the original blocks. Now, if after residual block $M < N$ the parameters of the residual terms are set to zero, i.e.,

$\mathbf{w}_{\mathcal{F},k} = 0$ for all $k = M, ..., N-1$, then all subsequent intermediate outputs and the final output are equal to the intermediate output $\hat{\mathbf{y}}_M$, i.e., $\hat{\mathbf{y}}_M$, $\hat{\mathbf{y}} = \hat{\mathbf{y}}_k = \hat{\mathbf{y}}_M$ for all $k = M, ..., N$. A formal result of this is provided in Lemma 2 in the Appendix. For a ResNet with only identity skip connections, the identity mapping also holds for the hidden states of the residual blocks, i.e., $\mathbf{w}_{\mathcal{F},k} = 0$ results in $\mathbf{x}_k = \mathbf{x}_{k+1}$, which, from a control perspective, corresponds to an equilibrium state. The parameters $\mathbf{w}_{\mathcal{E},k}$ of the intermediate outputs can be either shared with the backbone network (i.e., $\mathbf{w}_{\mathcal{E},k} = [\mathbf{w}_{\mathcal{S},k}, ..., \mathbf{w}_{\mathcal{S},N-1}, \mathbf{w}_{\mathcal{H}}]$) or can be additional parameters for increased representational capacity. This way, we can re-interpret or fine-tune ResNets trained without our intermediate outputs, since all necessary parameters are at hand.

Now, our optimal control problem implements the stage cost as the prediction loss of the intermediate output (2) with a weight $\gamma \geq 0$, where $\gamma = 0$ corresponds to the standard training,

$$\min_{\mathbf{w}} \; J_N(\mathbf{w}) = \sum_{k=0}^{N-1} \gamma \mathcal{L}(\hat{\mathbf{y}}_k) + \mathcal{L}(\hat{\mathbf{y}}) \quad \text{subject to} \quad (1), (2), \; \forall k = 0, ..., N-1. \tag{3}$$

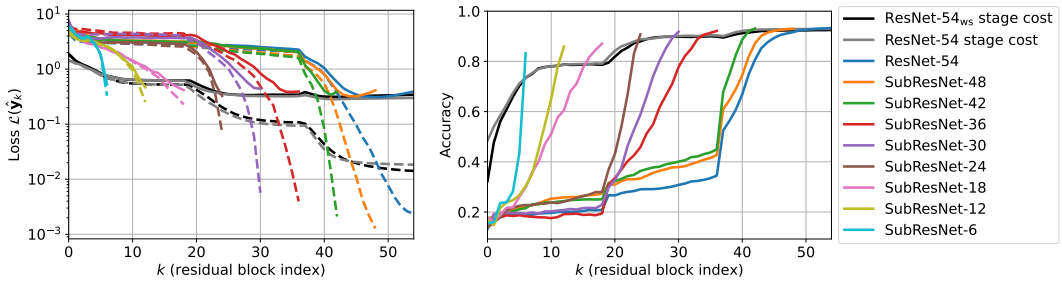
Next, we use this setup to derive bounds on the loss behavior of a deep ResNet trained with the stage cost loss by relating it to the loss of a shallower network sharing the architecture. To this end, we define a ResNet-M with M residual blocks and $M < N$ as a SubResNet architecture of a deeper ResNet-N with N residual blocks if the output of ResNet-M corresponds to the same architecture as the M-th intermediate output of ResNet-N. A formal definition is provided in Definition 1 in Appendix B. This implies that if for a ResNet-N and its SubResNet ResNet-M the shared parameters are equal, then the intermediate output of ResNet-N after M residual blocks $\hat{\mathbf{y}}_M^N$ will equal the output $\hat{\mathbf{y}}^M$ of the shallower ResNet-M, i.e., $\hat{\mathbf{y}}_M^N = \hat{\mathbf{y}}^M$. Moreover, if the additional parameters of ResNet-N, namely the parameters of its residual blocks $M, \ldots, N-1$, are set to zero, i.e., $\mathbf{w}_{\mathcal{F},k}^N = 0$ for $k = M, \ldots, N-1$, then all subsequent intermediate outputs of the deeper ResNet-N will produce the same prediction $\hat{\mathbf{y}}_k^N = \hat{\mathbf{y}}^N = \hat{\mathbf{y}}^M$, for all $k = M, \ldots, N$. This relationship is formalized in Lemma 3 in Appendix B. Building upon the identity mapping property of the ResNet, we now derive the following result on the asymptotic loss behavior of a ResNet trained with stage cost, based on the performance of a shallower SubResNet with traditional training.

Theorem 1 (ResNet asymptotics with weight decay) *Consider a ResNet-N with N residual blocks trained by (3) with stage cost weight $\gamma > 0$, and a weight decay $\frac{\lambda}{2}\|\mathbf{w}_{\mathcal{F},k}^N\|^2$ on the parameters on the residual terms $\mathbf{w}_{\mathcal{F},k}^N$. Let ResNet-M be its SubResNet (Definition 1), trained without the stage cost but the same weight decay. Let $\bar{\mathcal{L}} = \mathcal{L}(\hat{\mathbf{y}}^M)$ be the loss function of the output of ResNet-M. Then, the training objective for the optimal parameters $\mathbf{w}^N$ of ResNet-N satisfies the upper bound*

$$J_N(\mathbf{w}^N) = \sum_{k=0}^{N-1} \left[\gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) + \frac{\lambda}{2}\|\mathbf{w}_{\mathcal{F},k}^N\|^2 \right] + \mathcal{L}(\hat{\mathbf{y}}^N) \leq \bar{J}_N,$$

with $\bar{J}_N = \sum_{k=0}^{M-1} \left[\gamma \mathcal{L}(\hat{\mathbf{y}}_k^M) + \frac{\lambda}{2}\|\mathbf{w}_{\mathcal{F},k}^M\|^2 \right] + (1 + \gamma(N-M))\bar{\mathcal{L}}$ based on ResNet-M.

The proof is provided in Appendix B. If no weight decay is used, an adapted version of this bound yields $\mathcal{L}_{\text{avg}} = \frac{1}{N+1}\sum_{k=0}^{N} \mathcal{L}(\hat{\mathbf{y}}_k^N) \leq \bar{\mathcal{L}}_{\text{avg}} = \bar{\mathcal{L}} + \frac{C}{N+1}$, with $C < \infty$ based on the SubResNet-M. Moreover, if a loss function that attains its minimum at zero is used, e.g., the L2 loss, and assuming that there exists a SubResNet of depth N such that $\bar{\mathcal{L}} = 0$, then this bound tightens to $\mathcal{L}_{\text{avg}} \leq \frac{C}{N+1}$. This result is formalized in Theorem 4 in Appendix B.



(*a*) Loss. (*b*) Test accuracy.

Figure 2: Loss and accuracy trajectories for the standard ResNets trained on CIFAR-10 with the stage cost loss and their SubResNets of varying depth. The dashed loss lines are the training losses, and the solid lines are the test losses.

Table 1: Test accuracies (%) and the number of parameters (in millions M) for ResNets trained with the stage cost loss and standard training.

	Homogeneous ResNet			**Standard ResNet**
Model	**MNIST**	**CIFAR-10**	**CIFAR-100**	**CIFAR-10**
ResNet-54 standard training	99.64 (1.0 M)	93.05 (4.0 M)	71.94 (4.0 M)	93.28 (1.7 M)
SubResNet-12 standard training	99.63 (0.2 M)	91.43 (0.9 M)	68.77 (0.9 M)	85.93 (0.06 M)
ResNet-54$_{ws}$ stage cost	99.62 (1.0 M)	91.59 (4.0 M)	69.97 (4.0 M)	92.51 (1.7 M)
SubResNet-12 pruned	99.54 (0.2 M)	91.02 (0.9 M)	66.55 (0.9 M)	78.74 (0.06 M)
ResNet-54 stage cost	99.68 (1.0 M)	91.64 (4.0 M)	69.18 (4.4 M)	92.91 (1.9 M)
SubResNet-12 pruned	99.64 (0.2 M)	91.26 (0.9 M)	66.66 (0.9 M)	78.64 (0.06 M)

4. Experiments

In our experiments, we train ResNets on MNIST [8], CIFAR-10, and CIFAR-100 [7] using the proposed stage cost loss and compare their loss and accuracy trajectories to those of ResNets of varying depth trained with standard training. The results for CIFAR-10 are shown in Figure 2 and the implementation details are provided in Appendix A.1.

Results. Our empirical evaluation shows that the bounds provided in 1 and 4 are relatively tight, see Appendix B. The loss trajectories shown in Figure 2 (b) show that ResNets trained with the proposed stage cost already achieve a good fit after 12 residual blocks, reaching 78.74% test accuracy for the ResNet$_{\text{WS}}$ in which the parameters of the intermediate outputs are shared. Performance plateaus for the rest of the first stage of layers with the same number of filters and improves at the onset of subsequent stages with increasing representational capacity. In contrast, networks trained with standard training only show a good fit at their final output, but with higher test accuracy, especially in the first two stages. Appendix A.2 provides an in-depth analysis of intermediate output convergence and parameter dynamics, along with results from training a homogeneous ResNet with a fixed number of filters. Interestingly, our stage cost loss formulation leads to a principled way of identifying layers that could be pruned due to insignificant residual contribution. Table 1 compares test accuracies of pruned SubResNet-12 to standard training, showing pruned homogeneous models differ only up to 3.5% points, while it remains challenging to prune standard ResNets nearly lossless.

Outlook. Our proposed training scheme offers promising future directions. The post-hoc pruning can be understood as finding the optimal SubResNet, computing the best representations for a downstream task. When considering varying dimensionality across residual blocks, our current analysis does not lead to straightforward pruning like in the simpler homogeneous models. In the future, we want to investigate the impact of the observed stage-wise dynamics, possibly leading us to tighter bounds and eventually surgical pruning of intermediate layers.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Research, Technology, and Space (BMFTR) via the 6GEM research hub (16KISK038), and by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) via the TRR 391 Spatio-temporal Statistics for the Transition of Energy and Transport (project 520388526) and the research unit Active Learning for Systems and Control (project 535860958).

References

[1] Bo Chang, Lili Meng, Eldad Haber, Lars Ruthotto, David Begert, and Elliot Holtham. Reversible architectures for arbitrarily deep residual neural networks. In *The 32th Annual AAAI Conference on Artificial Antelligence*, 2018.

[2] Carlos Esteve and Borjan Geshkovski. Sparsity in long-time control of neural odes. *Systems & Control Letters*, 172:105452, 2023.

[3] Carlos Esteve, Borjan Geshkovski, Dario Pighin, and Enrique Zuazua. Large-time asymptotics in deep learning. *arXiv preprint arXiv:2008.02491*, 2020.

[4] Timm Faulwasser, Arne-Jens Hempel, and Stefan Streif. On the turnpike to design of deep neural networks: Explicit depth bounds. *IFAC Journal of Systems and Control*, 30:100290, 2024.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. Technical Report.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[9] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-Supervised Nets. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[10] Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. Split computing and early exiting for deep learning applications: Survey and research challenges. *ACM Computing Surveys*, 55(5), December 2022.

[11] Keyan Miao and Konstantinos Gatsis. How deep do we need: Accelerating training and inference of neural ODEs via control perspective. In *International Conference on Machine Learning (ICML)*, 2024.

[12] Jens Püttschneider and Timm Faulwasser. On dissipativity of cross-entropy loss in training resnets. *arXiv preprint arXiv:2405.19013*, 2024.

[13] Haseena Rahmath P, Vishal Srivastava, Kuldeep Chaurasia, Roberto G. Pacheco, and Rodrigo S. Couto. Early-exit deep neural network - a comprehensive survey. *ACM Computing Surveys*, 57 (3), November 2024.

Appendix A. Experimental Supplements

A.1. Setup

The experimental setup is based on [5]. All of our experiments used a pre-activation ResNet with 54 residual blocks, with two convolutional layers per residual block. In contrast to our notation, [5] defines the depth of a ResNet as the total number of layers, including convolutional and fully connected layers. Hence, the ResNet-54 in this paper corresponds to the ResNet-110 in [5]. The standard ResNet architecture consists of an input embedding with of a 3x3 convolution with 16 filters followed by a batch normalization and ReLU. This is followed by 54 residual blocks organized in three stages, each with 18 residual blocks. The convolutional layers within these stages use 16, 32, and 64 filters, respectively. The output block consists of a batch normalization layer, a ReLU activation, global average pooling, and a final linear layer that predicts the logits. Similar to [5], we train for 165 epochs with a batch size of 128 using SGD with an initial learning rate of $\alpha = 0.1$ and a momentum of $\mu = 0.9$, after 82 epochs, the learning rate is divided by ten and again after 123 epochs. A constant stage cost weight of $\gamma = 0.02$ is used for all experiments.

For ResNets trained without the stage cost, we reuse the backbone network parameters to compute the trajectory of the intermediate outputs. When training with the stage cost, we consider two variants: $\text{ResNet}_{\text{WS}}$, which shares parameters between the backbone and the intermediate outputs, and the full stage-cost ResNet, which introduces additional parameters for the intermediate outputs. For the additional parameters associated with the intermediate outputs, the learning rate is set to α/γ to ensure that their effective learning rate matches that of the final output layer. This latter approach increases the total number of parameters during training. However, when pruning, these additional intermediate output blocks and their parameters are removed, such that all models have the same number of parameters. In our training approach, the output block is applied to all hidden states. For each of these states use a separate batch norm statistics, while sharing the batch norm weights and bias for the $\text{ResNet}_{\text{WS}}$.

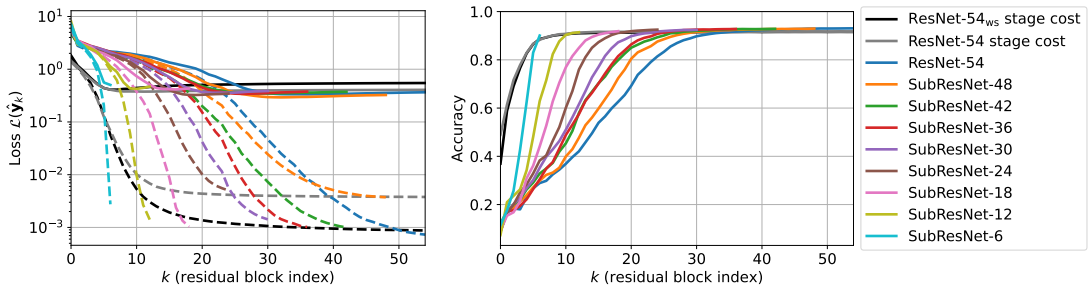
The homogeneous ResNets have a constant number of convolutional filters, 64 CIFAR and 32 for MNIST, in all residual blocks and the input embedding. Apart from this modification, their architecture is identical to the standard ResNet.

A.2. Additional Experiments

For the standard ResNet shown in Figure 2, we observe that accuracy and loss trajectories improve noticeably at the beginning of each stage. This effect is likely due to the increased representational capacity resulting from the doubling of convolutional filters from one stage to the next. To verify this, we consider a homogeneous ResNet, in which all residual blocks use a constant number of filters, fixed at 64. The resulting loss and accuracy trajectories for CIFAR-10 are shown in Figure 3. In this homogeneous architecture, convergence is stronger compared to the standard ResNet. After approximately 12 residual blocks, the loss and accuracy trajectories stabilize, and no further improvement is observed. This suggests that the remaining residual blocks can be pruned without significant performance loss. The resulting SubResNet-12 achieves a test accuracy of 91.26%, compared to the 91.64% of the full ResNet-54, which is close to the 91.43% obtained by training a SubResNet-12 by standard training (see Table 1). Hence, the proposed training formulation helps to determine the depth suitable for a given task. The same convergence behavior is observed for the homogeneous ResNets trained on MNIST and CIFAR-100. To close the performance gap after pruning, we plan to invest future research into fine-tuning for a small number of epochs.

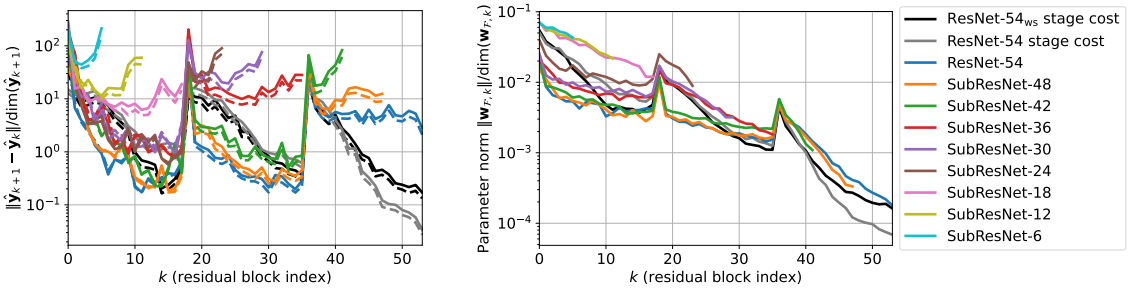
Notably, our ablation using weight-sharing in the intermediate outputs, denoted as $\text{ResNet}_{\text{WS}}$, revealed that a ResNet with additional parameters only achieved insignificant accuracy gains. This indicates that our stage-cost trained weight-sharing $\text{ResNet}_{\text{WS}}$ can be used to achieve comparable performance while using a reduced number of parameters.

Moreover, we can study the convergence in the residuals between consecutive intermediate outputs $\hat{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_k$ and the norm of the parameters are shown in Figure 4 for the standard ResNet and in Figure 5 for the homogeneous ResNet (using 64 filters across layers). We observe in Figure 4 (b) and Figure 5 (b) that while the parameter norms converge for all ResNet variants, the parameters of the ResNet-54 trained with the stage cost converge faster than those of the same model trained without it. Especially in the homogeneous case, in Figure 5 (a), the residual contribution of subsequent blocks vanishes when training with our stage cost loss. This underscores the ability to discard them from forward propagation entirely. When considering the standard setup in Figure 4 (a), this implication holds only for stages operating in the same representational space, in particular for the last stage. We plan to conduct more research on this effect to eventually be able to surgically prune blocks.



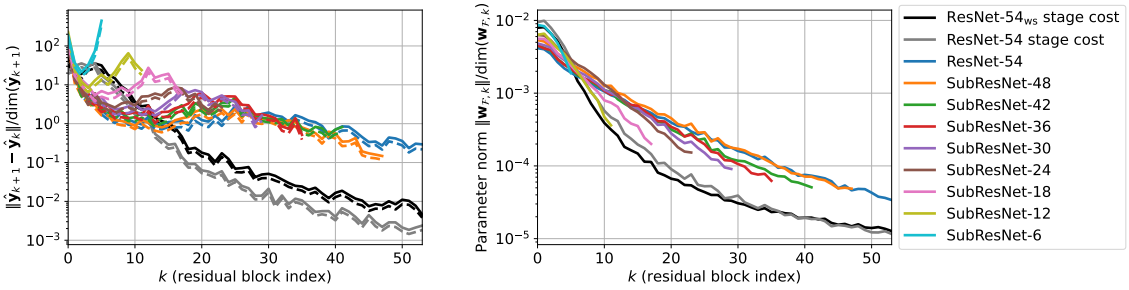
(*a*) Loss. (*b*) Test accuracy.

Figure 3: Loss and accuracy trajectories for the ResNets with homogeneous architecture trained on CIFAR-10 with the stage cost loss and their SubResNets of varying depth. The dashed loss lines represent are the training losses and the solid lines are the test losses.



(a) Difference norms (b) Parameter norms.

Figure 4: Output residual norms and parameter norm residual trajectories for the standard ResNets trained on CIFAR-10 with the stage cost loss and their SubResNets of varying depth. The dashed loss lines represent are the training losses and the solid lines are the test losses.



(a) Difference norms (b) Parameter norms.

Figure 5: Output residual norms and parameter norm residual trajectories for the homogeneous ResNets trained on CIFAR-10 with the stage cost loss and their SubResNets of varying depth. The dashed loss lines represent are the training losses and the solid lines are the test losses.

Appendix B. Asymptotic Loss Bounds

In this appendix, we provide the formal results corresponding to Section 3. To prove Theorem 1, we build a rigorous setup with intermediate results. The main idea is based on the observation that with our proposed intermediate outputs (2), the ResNet can learn an identity mapping between subsequent early exits.

Lemma 2 (Identity mapping) *Consider the ResNet (1), with the intermediate outputs chosen according to (2). Let the intermediate outputs of residual block M produce the prediction $\hat{\mathbf{y}}_M$, and let the parameters of all following residual terms be zero, $\mathbf{w}_{\mathcal{F},k} = 0$, $k = M, ..., N-1$. Then all subsequent intermediate outputs and the final output are equal*

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_k = \hat{\mathbf{y}}_M, \tag{4}$$

for all $k = M, ..., N$.

Proof If $\mathbf{w}_{\mathcal{F},k} = 0$, for the residual blocks $k = M, ..., N-1$, then $\mathcal{F}_k(\mathbf{x}_k, 0) = 0$ for each of these blocks. Hence, the state is being propagated by $\mathbf{x}_{k+1} = \mathcal{S}_k(\mathbf{x}_k, \mathbf{w}_{\mathcal{S},k})$ for all $k = M, ..., N-1$. This gives the output $\hat{\mathbf{y}} = \mathcal{H}(\mathbf{x}_N, \mathbf{w}_{\mathcal{H}})$ and the intermediate outputs (2), which corresponds to (4). ■

Definition 1 (SubResNet architecture) *We refer to the architecture of ResNet-M with M residual blocks* [2] *as a SubResNet architecture of ResNet-N with N residual blocks, $N > M$, if the architecture of the output $\mathcal{H}^M$ of ResNet-M is shared with the architecture of the intermediate output $\mathcal{E}_M^N$ of ResNet-N. That is, if all residual blocks of ResNet-M are contained in ResNet-N, i.e. $f_k^M = f_k^N$ for all $k = 0, ..., M-1$. Moreover, if ResNet-N has non identity skip connections in residual blocks $k \geq M$, then those need to be included in the output of ResNet-M, i.e. $\mathcal{H}^M(x, \mathbf{w}_{\mathcal{H}}^M) = \mathcal{E}_k^N(x, \mathbf{w}_{\mathcal{E},k}^N)$, with $\mathbf{w}_{\mathcal{H}}^M = \left[\mathbf{w}_{\mathcal{S},M}^N, ..., \mathbf{w}_{\mathcal{S},N-1}^N, \mathbf{w}_{\mathcal{H}}^N\right]$.*

This definition of the SubResNet architecture implies that the identity mapping (Lemma 2) also holds between a ResNet and its SubResNets.

Lemma 3 (Identity mapping for SubResNets) *Consider ResNet-N with N residual blocks and ResNet-M with a SubResNet architecture (Definition 1), consisting of M residual blocks and producing the output $\hat{\mathbf{y}}^M$. Let the parameters $\bar{\mathbf{w}}^N$ of the first M residual blocks of ResNet-N be equal to those of ResNet-M, i.e. $\bar{\mathbf{w}}_k^N = \mathbf{w}_k^M$ $k = 0, ..., M-1$, and choose the parameters of following residual terms equal to zero, $\bar{\mathbf{w}}_{\mathcal{F},k}^N = 0$ for $k = M, ..., N-1$. Then, all intermediate outputs deeper than M and the output of ResNet-N are equal to the output of ResNet-M, i.e., $\hat{\mathbf{y}}_k^N = \hat{\mathbf{y}}^N = \hat{\mathbf{y}}^M$ for all $k = M, ..., N$.*

The proof follows directly from Lemma 2 and is thus omitted. We are now ready to prove our main result, which is repeated for the ease of readability.

Theorem 1 (ResNet asymptotics with weight decay) *Consider a ResNet-N with N residual blocks trained by (3) with stage cost weight $\gamma > 0$, and a weight decay $\frac{\lambda}{2}\|\mathbf{w}_{\mathcal{F},k}^N\|^2$ on the parameters on the residual terms $\mathbf{w}_{\mathcal{F},k}^N$. Let ResNet-M be its SubResNet (Definition 1), trained without the stage cost but the same weight decay. Let $\bar{\mathcal{L}} = \mathcal{L}(\hat{\mathbf{y}}^M)$ be the loss function of the output of ResNet-M. Then, the training objective for the optimal parameters $\mathbf{w}^N$ ResNet-N satisfies the upper bound*

$$J_N(\mathbf{w}^N) = \sum_{k=0}^{N-1} \left[\gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) + \frac{\lambda}{2}\|\mathbf{w}_{\mathcal{F},k}^N\|^2\right] + \mathcal{L}(\hat{\mathbf{y}}^N) \leq \bar{J}_N,$$

with $\bar{J}_N = \sum_{k=0}^{M-1} \left[\gamma \mathcal{L}(\hat{\mathbf{y}}_k^M) + \frac{\lambda}{2}\|\mathbf{w}_{\mathcal{F},k}^M\|^2\right] + (1 + \gamma(N-M))\bar{\mathcal{L}}$ based on the objective of ResNet-M.

Proof Let $\hat{\mathbf{y}}_k^M$ denote the trajectory of the intermediate outputs of the ResNet-M. Then by Lemma 3, the feasible parameters $\bar{\mathbf{w}}^N$ of for ResNet-N, with $\bar{\mathbf{w}}_k^N = \mathbf{w}_k^M$ $k = 0, ..., M-1$ and $\bar{\mathbf{w}}_{\mathcal{F},k}^N = 0$ for $k = N, ..., M-1$, produce the trajectory of intermediate outputs

$$\bar{\mathbf{y}}_k^N = \begin{cases} \hat{\mathbf{y}}_k^M & \text{if } k < M \\ \hat{\mathbf{y}}_M^M & \text{if } k \geq M, \end{cases}$$

2. For simplicity, we refer to the network architecture for $M = 0$, which only consists of the input stem and the output layer, as a ResNet, despite its lack of skip connections.

for all $k = 0, ..., N$. The feasible parameters $\bar{\mathbf{w}}^N$ induce in the objective with stage cost loss

$$\bar{J}_N = J_N(\bar{\mathbf{w}}^N) = \sum_{k=0}^{M-1} \left[\gamma \mathcal{L}(\hat{\mathbf{y}}_k^M) + \frac{\lambda}{2} \|\bar{\mathbf{w}}_{\mathcal{F},k}^M\|^2 \right] + (\gamma(N-M)+1)\bar{\mathcal{L}}.$$

Which is an upper bound for the optimal intermediate outputs $\hat{\mathbf{y}}_k^N$ and parameters $\mathbf{w}^N$ obtained from training with the stage cost loss

$$J_N(\mathbf{w}^N) = \sum_{k=0}^{N-1} \left[\gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) + \frac{\lambda}{2} \|\mathbf{w}_{\mathcal{F},k}^N\|^2 \right] + \mathcal{L}(\hat{\mathbf{y}}_N^N) \leq \bar{J}_N.$$

■

The second bound is an extension of Theorem 1 for the training with stage cost loss in the absence of weight decay.

Theorem 4 (ResNet asymptotics without weight decay) *Consider a ResNet-N trained with stage cost, $\gamma > 0$, and N residual blocks and its SubResNet architecture (Definition 1) ResNet-M trained without stage cost. Let $\bar{\mathcal{L}} = \mathcal{L}(\hat{\mathbf{y}}^M)$ be the loss function of the output of ResNet-M. Then, the average loss along the intermediate outputs $\mathcal{L}_{\text{avg}}$ of ResNet-N satisfies*

$$\mathcal{L}_{\text{avg}} = \frac{1}{N+1} \sum_{k=0}^{N} \mathcal{L}(\hat{\mathbf{y}}_k^N) \leq \bar{\mathcal{L}}_{\text{avg}} = \bar{\mathcal{L}} + \frac{C}{N+1}, \tag{5}$$

for $C = \sum_{k=0}^{M-1} \mathcal{L}(\hat{\mathbf{y}}_k^M) + \frac{1-\gamma-\gamma M}{\gamma} \bar{\mathcal{L}}$ relating to the performance of the shallower network ResNet-M.

Proof Let $\hat{\mathbf{y}}_k^M$ denote the trajectory of intermediate outputs of ResNet-M. Then by Lemma 3, choosing the parameters of for ResNet-N $\bar{\mathbf{w}}^N$ as $\bar{\mathbf{w}}_k^N = \mathbf{w}_k^M$ $k = 0, ..., M-1$ and $\bar{\mathbf{w}}_{\mathcal{F},k}^N = 0$ for $k = M, ..., N-1$ produce the trajectory of intermediate outputs

$$\bar{\mathbf{y}}_k^N = \begin{cases} \hat{\mathbf{y}}_k^M & \text{if } k < M \\ \hat{\mathbf{y}}_N^M & \text{if } k \geq N, \end{cases}$$

for $k = 0, ..., N-1$. This feasible trajectory of parameters $\bar{\mathbf{w}}^N$ and intermediate outputs $\bar{\mathbf{y}}^N$ induces in the objective with stage cost

$$J_N(\bar{\mathbf{w}}^N) = \sum_{k=0}^{N-1} \gamma \mathcal{L}(\bar{\mathbf{y}}_k^N) + \mathcal{L}(\bar{\mathbf{y}}_N^N) = \sum_{k=0}^{M-1} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^M) + (\gamma(N-M)+1)\bar{\mathcal{L}}.$$

This is an upper bound to training objective of the trained ResNet-N trained with the stage cost loss with the optimal parameters $\mathbf{w}^N$ and intermediate outputs $\hat{\mathbf{y}}^M$

$$J_N(\mathbf{w}^N) = \sum_{k=0}^{N-1} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) + \mathcal{L}(\hat{\mathbf{y}}_N^N) \leq J_N(\bar{\mathbf{w}}^N) = \sum_{k=0}^{M-1} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^M) + (1+\gamma(N-M))\bar{\mathcal{L}}.$$

Using that $\gamma \leq 1$, we obtain

$$\sum_{k=0}^{N} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) \leq \sum_{k=0}^{M-1} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^M) + (1 + \gamma(N - M))\bar{\mathcal{L}}.$$

With $\bar{\mathcal{L}} = \gamma\bar{\mathcal{L}} + (1 - \gamma)\bar{\mathcal{L}}$ and by rearranging we obtain

$$\sum_{k=0}^{N} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) \leq \sum_{k=0}^{M-1} \gamma \mathcal{L}(\hat{\mathbf{y}}_k^N) + (1 - \gamma - \gamma M)\bar{\mathcal{L}} + \gamma(N + 1)\bar{\mathcal{L}}. \tag{6}$$

Dividing by $\gamma(N + 1)$ gives (5) with $C = \sum_{k=0}^{M-1} \mathcal{L}(\hat{\mathbf{y}}_k^M) + \frac{1-\gamma-\gamma M}{\gamma}\bar{\mathcal{L}}$. ■

This implies that asymptotically, for $N \to \infty$, the average loss of of the ResNet trained with the stage cost loss will be equal to that of the shallower SubResNet.

However, we emphasize that these bounds focus solely on the training dynamics, and no guarantees are provided regarding validation performance. Our theoretical results require that the empirical loss is optimized to global optimality. Moreover, if a loss function that attains its minimum at zero is used, e.g., the L2 loss, and assuming that there exists a SubResNet of depth M such that $\bar{\mathcal{L}} = 0$, then this bound tightens to $\mathcal{L}_{\text{avg}} \leq \frac{C}{N+1}$.

The results of the bounds for CIFAR-10 are shown in Table 2 for the standard ResNet and in Table 3 for the homogeneous ResNet. For the homogeneous ResNet, we find Bound 1 on the training objective with weight decay (Theorem 1) to be moderately tight. The bound obtained from the SubResNet of depth 6 is closest, since it is the shallowest network that achieves a sufficiently low training loss. The SubResNet of depth 0, which cannot fit the dataset properly, leads to a looser depth bound due to its larger value of $\bar{\mathcal{L}}$. For deeper SubResNets, the summation of the higher loss function values of the intermediate outputs leads to an overly conservative bound. We evaluate the Bound 2 from Theorem 4 for the training without weight decay for the trajectories obtained by training with small weight decay of $\lambda = 10^{-4}$. The results on the average loss function are less tight due to the higher loss values for the outputs of the first residual blocks of the ResNet trained by standard training. In contrast to Bound 1, these are not weighted by the stage cost loss weight γ.

In general, we find that for the standard ResNet, our bounds are less tight. One reason is this difference between the representational abilities of the full ResNet and the shallower SubResNets, which only contain residual blocks with 16 and 32 convolution filters.

Table 2: Results of loss bounds for the standard ResNet trained on CIFAR-10.

SubResNet depth M	$\bar{\mathcal{L}}$	**Bound 1** (Theorem 1)			**Bound 2** (Theorem 4)		
		$J_M(w^M)$	$\bar{J}_M$	$\bar{J}_M - J_M(w^M)$	$\mathcal{L}_{\text{avg}}$	$\bar{\mathcal{L}}_{\text{avg}}$	$\bar{\mathcal{L}}_{\text{avg}} - \mathcal{L}_{\text{avg}}$
0	1.326	0.570	2.758	2.187	0.340	2.507	2.167
6	0.327	0.570	1.173	0.603	0.340	21.238	20.898
12	0.252	0.570	1.186	0.616	0.340	28.063	27.723
18	0.228	0.570	1.274	0.704	0.340	34.451	34.112
24	0.072	0.570	1.549	0.979	0.340	58.629	58.289
30	0.006	0.570	2.128	1.558	0.340	90.931	90.591
36	0.004	0.570	2.409	1.839	0.340	103.758	103.418
42	0.002	0.570	2.189	1.618	0.340	93.293	92.953
48	0.001	0.570	2.253	1.682	0.340	96.334	95.994
54	0.002	0.570	2.232	1.662	0.340	94.822	94.483

Table 3: Results of loss bounds for the homogeneous ResNet trained on CIFAR-10.

SubResNet depth M	$\bar{\mathcal{L}}$	**Bound 1** Theorem 1)			**Bound 2** (Theorem 4)		
		$J_M(w^M)$	$\bar{J}_M$	$\bar{J}_M - J_M(w^M)$	$\mathcal{L}_{\text{avg}}$	$\bar{\mathcal{L}}_{\text{avg}}$	$\bar{\mathcal{L}}_{\text{avg}} - \mathcal{L}_{\text{avg}}$
0	1.188	0.245	2.470	2.225	0.087	2.246	2.159
6	0.003	0.245	0.421	0.176	0.087	13.275	13.188
12	0.001	0.245	0.537	0.292	0.087	18.518	18.431
18	0.001	0.245	0.574	0.330	0.087	20.260	20.173
24	0.005	0.245	0.728	0.483	0.087	25.312	25.225
30	0.001	0.245	0.761	0.516	0.087	28.229	28.142
36	0.001	0.245	0.813	0.568	0.087	30.843	30.756
42	0.001	0.245	0.801	0.556	0.087	30.287	30.200
48	0.004	0.245	0.925	0.680	0.087	34.440	34.353
54	0.001	0.245	0.942	0.697	0.087	36.772	36.685