Appendix: Learning Deep ResNet Blocks Sequentially using Boosting Theory

A. Related Works

A.1. Loss function and architecture selection

In neural network optimization, there are many commonly-used loss functions and criteria, e.g., mean squared error, negative log likelihood, margin criterion, etc. There are extensive works (Girshick, 2015; Rubinstein & Kroese, 2013; Tygert et al., 2015) on selecting or modifying loss functions to prevent empirical difficulties such as exploding/vanishing gradients or slow learning (Balduzzi et al., 2017). However, there are no rigorous principles for selecting a loss function in general. Other works consider variations of the multilayer perceptron (MLP) or convolutional neural network (CNN) by adding identity skip connections (He et al., 2016), allowing information to bypass particular layers. However, no theoretical guarantees on the training error are provided despite breakthrough empirical successes. Hardt et al. (Hardt & Ma, 2016) have shown the advantage of identity loops in linear neural networks with theoretical justifications; however the linear setting is unrealistic in practice.

A.2. Learning algorithm design

There have been extensive works on improving BP (LeCun et al., 1989). For instance, momentum (Qian, 1999), Nesterov accelerated gradient (Nesterov, 1983), Adagrad (Duchi et al., 2011) and its extension Adadelta (Zeiler, 2012). Most recently, Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014), a combination of momentum and Adagrad, has received substantial success in practice. All these methods are modifications of stochastic gradient descent (SGD), but our method only requires an arbitrary oracle, which does not necessarily need to be an SGD solver, that solves a relatively simple shallow neural network.

B. Proof for Lemma 3.2: the strong learner is a ResNet

Proof. In our algorithm, the input of the next module is the output of the current module

$$g_{t+1}(x) = f_t(g_t(x)) + g_t(x), \tag{8}$$

we thus obtain that each weak learning module is

$$\begin{aligned} h_t(x) &= \alpha_{t+1}\mathbf{w}_{t+1}^\top(f_t(g_t(x)) + g_t(x)) - \alpha_t\mathbf{w}_t^\top g_t(x) &(9)\\ &= \alpha_{t+1}\mathbf{w}_{t+1}^\top g_{t+1}(x) - \alpha_t\mathbf{w}_t^\top g_t(x), &(10)\end{aligned}$$

and similarly

$$h_{t+1} = \alpha_{t+2}\mathbf{w}_{t+2}^\top g_{t+2}(x) - \alpha_{t+1}\mathbf{w}_{t+1}^\top g_{t+1}(x). \tag{11}$$

Therefore the sum over $h_t(x)$ and $h_{t+1}(x)$ is

$$h_t(x) + h_{t+1}(x) = \alpha_{t+2}\mathbf{w}_{t+2}^\top g_{t+2}(x) - \alpha_t\mathbf{w}_t^\top g_t(x) \tag{12}$$

And we further see that the weighted summation over all $h_t(x)$ is a telescoping sum

$$\sum_{t=1}^{T} h_t(x) = \alpha_{T+1}w_{T+1}^\top g_{T+1}(x) - \alpha_1 w_1^\top g_1(x) = \alpha_{T+1}w_{T+1}^\top g_{T+1}(x). \tag{13}$$

□

C. Proof for Theorem 4.2: binary class telescoping sum boosting theory

Proof. We will use a 0-1 loss to measure the training error. In our analysis, the 0-1 loss is bounded by exponential loss.

The training error is therefore bounded by

$$\Pr_{i\sim D_1}(p(\alpha_{T+1}w_{T+1}^\top g_{T+1}(x_i)) \neq y_i) \tag{14}$$

$$= \sum_{i=1}^m D_1(i)\mathbf{1}\{\tilde{\sigma}(\alpha_{T+1}w_{T+1}^\top g_{T+1}(x_i)) \neq y_i\} \tag{15}$$

$$= \sum_{i=1}^m D_1(i)\mathbf{1}\left\{\tilde{\sigma}\left(\sum_{t=1}^T h_t(x_i)\right) \neq y_i\right\} \tag{16}$$

$$\leq \sum_{i=1}^m D_1(i)\exp\left\{-y_i\sum_{t=1}^T h_t(x_i)\right\} \tag{17}$$

$$= \sum_{i=1}^m D_{T+1}(i)\prod_{t=1}^T Z_t \tag{18}$$

$$= \prod_{t=1}^T Z_t \tag{19}$$

where $Z_t = \sum_{i=1}^m D_t(i)\exp\left(-y_i h_t(x_i)\right)$.

We choose α_{t+1} to minimize Z_t.

$$\frac{\partial Z_t}{\partial \alpha_{t+1}} = -\sum_{i=1}^m D_t(i)y_i o_{t+1}\exp\left(-y_i h_t(x_i)\right) \tag{20}$$

$$= -Z_t\sum_{i=1}^m D_{t+1}(i)y_i o_{t+1}(i) = 0 \tag{21}$$

Furthermore each learning module is bounded as we see in the following analysis. We obtain

$$Z_t = \sum_{i=1}^m D_t(i)e^{-y_i h_t(x_i)} \tag{22}$$

$$= \sum_{i=1}^m D_t(i)e^{-\alpha_{t+1}y_i o_{t+1}(x_i)+\alpha_t y_i o_t(x_i)} \tag{23}$$

$$\leq \sum_{i=1}^m D_t(i)e^{-\alpha_{t+1}y_i o_{t+1}(x_i)}\sum_{i=1}^m D_t(i)e^{\alpha_t y_i o_t(x_i)} \tag{24}$$

$$= \sum_{i=1}^m D_t(i)e^{-\alpha_{t+1}\frac{1+y_i o_{t+1}(x_i)}{2}+\alpha_{t+1}\frac{1-y_i o_{t+1}(x_i)}{2}}\sum_{i=1}^m D_t(i)e^{\alpha_t\frac{1+y_i o_t(x_i)}{2}-\alpha_t\frac{1-y_i o_t(x_i)}{2}} \tag{25}$$

$$\leq \sum_{i=1}^m D_t(i)\left(\frac{1+y_i o_{t+1}(x_i)}{2}e^{-\alpha_{t+1}}+\frac{1-y_i o_{t+1}(x_i)}{2}e^{\alpha_{t+1}}\right)\cdot\sum_{i=1}^m D_t(i)\left(\frac{1+y_i o_t(x_i)}{2}e^{\alpha_t}+\frac{1-y_i o_t(x_i)}{2}e^{-\alpha_t}\right) \tag{26}$$

$$= \sum_{i=1}^m D_t(i)\left(\frac{1+y_i o_{t+1}(x_i)}{2}e^{-\alpha_{t+1}}+\frac{1-y_i o_{t+1}(x_i)}{2}e^{\alpha_{t+1}}\right)\frac{e^{\alpha_t}+e^{-\alpha_t}}{2} \tag{27}$$

$$= \sum_{i=1}^m D_t(i)\left(\frac{e^{-\alpha_{t+1}}+e^{\alpha_{t+1}}}{2}+\frac{e^{-\alpha_{t+1}}-e^{\alpha_{t+1}}}{2}y_i o_{t+1}(x_i)\right)\frac{e^{\alpha_t}+e^{-\alpha_t}}{2} \tag{28}$$

$$= \left(\frac{e^{-\alpha_{t+1}}+e^{\alpha_{t+1}}}{2}+\frac{e^{-\alpha_{t+1}}-e^{\alpha_{t+1}}}{2}\tilde{\gamma}_t\right)\frac{e^{\alpha_t}+e^{-\alpha_t}}{2} \tag{29}$$

Equation (24) is due to the non-positive correlation between $\exp(-yo_{t+1}(x))$ and $\exp(yo_t(x))$. Jensen's inequality in Equation (26) holds only when $|y_i o_{t+1}(x_i)| \leq 1$ which is satisfied by the definition of the weak learning module.

The algorithm chooses α_{t+1} to minimize Z_t. We achieve an upper bound on Z_t, $\sqrt{\frac{1-\tilde{\gamma}_t^2}{1-\tilde{\gamma}_{t-1}^2}}$ by minimizing the bound in Equation (29)

$$Z_t|_{\alpha_{t+1}=\arg\min Z_t} \leq Z_t|_{\alpha_{t+1}=\frac{1}{2}\ln\left(\frac{1+\tilde{\gamma}_t}{1-\tilde{\gamma}_t}\right)} \tag{30}$$

$$\leq \left(\frac{e^{-\alpha_{t+1}}+e^{\alpha_{t+1}}}{2}+\frac{e^{-\alpha_{t+1}}-e^{\alpha_{t+1}}}{2}\tilde{\gamma}_t\right)\frac{e^{\alpha_t}+e^{-\alpha_t}}{2}\Bigg|_{\alpha_{t+1}=\frac{1}{2}\ln\left(\frac{1+\tilde{\gamma}_t}{1-\tilde{\gamma}_t}\right)} \tag{31}$$

$$= \sqrt{\frac{1-\tilde{\gamma}_t^2}{1-\tilde{\gamma}_{t-1}^2}} = \sqrt{1-\gamma_t^2} \tag{32}$$

Therefore over the T modules, the training error is upper bounded as follows

$$\Pr_{i\sim D}\left(p(\alpha_{T+1}w_{T+1}^\top g_{T+1}(x_i))\right) \neq y_i) \leq \prod_{t=1}^{T}\sqrt{1-\gamma_t^2} \leq \prod_{t=1}^{T}\sqrt{1-\gamma^2} = \exp\left(-\frac{1}{2}T\gamma^2\right) \tag{33}$$

Overall, Algorithm 1 leads us to consistent learning of ResNet. □

D. Proof for Corollary 4.3: Generalization Bound

Rademacher complexity technique is powerful for measuring the complexity of $\mathcal{H}$ any family of functions $h : \mathcal{X} \to \mathbb{R}$, based on easiness of fitting any dataset using classifiers in $\mathcal{H}$ (where $\mathcal{X}$ is any space). Let $S =< x_1, \ldots, x_m >$ be a sample of m points in $\mathcal{X}$. The empirical Rademacher complexity of $\mathcal{H}$ with respect to S is defined to be

$$\mathcal{R}_S(\mathcal{H}) \stackrel{\text{def}}{=} \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\right] \tag{34}$$

where σ is the Rademacher variable. The Rademacher complexity on m data points drawn from distribution $\mathcal{D}$ is defined by

$$\mathcal{R}_m(\mathcal{H}) = \mathbb{E}_{S\sim\mathcal{D}}\left[\mathcal{R}_S(\mathcal{H})\right]. \tag{35}$$

Proposition D.1. *(Theorem 1 (Cortes et al., 2014)) Let $\mathcal{H}$ be a hypothesis set admitting a decomposition $\mathcal{H} = \cup_{i=1}^{l}\mathcal{H}_i$ for some $l > 1$. $\mathcal{H}_i$ are distinct hypothesis sets. Let S be a random sequence of m points chosen independently from $\mathcal{X}$ according to some distribution $\mathcal{D}$. For $\theta > 0$ and any $H = \sum_{t=1}^{T} h_t$, with probability at least $1-\delta$,*

$$\Pr_{\mathcal{D}}\left(yH(x) \leq 0\right) \leq \Pr_{S}\left(yH(x) \leq \theta\right) + \frac{4}{\theta}\sum_{t=1}^{T}\mathcal{R}_m(\mathcal{H}_{k_t}) + \frac{2}{\theta}\sqrt{\frac{\log l}{m}}$$
$$+\sqrt{\lceil\frac{4}{\theta^2}\log\left(\frac{\theta^2 m}{\log l}\right)\rceil\frac{\log l}{m}+\frac{\log\frac{2}{\delta}}{2m}} \tag{36}$$

for all $h_t \in \mathcal{H}_{k_t}$.

Lemma D.2. *Let $\tilde{h} = \tilde{\mathbf{w}}^\top\tilde{\mathbf{f}}$, where $\tilde{\mathbf{w}} \in \mathbb{R}^n$, $\tilde{\mathbf{f}} \in \mathbb{R}^n$. Let $\tilde{\mathcal{H}}$ and $\tilde{\mathcal{F}}$ be two hypothesis sets, and $\tilde{h} \in \tilde{\mathcal{H}}$, $\tilde{\mathbf{f}}_j \in \tilde{\mathcal{F}}$, $\forall j \in [n]$. The Rademacher complexity of $\tilde{\mathcal{H}}$ and $\tilde{\mathcal{F}}$ with respect to m points from $\mathcal{D}$ are related as follows*

$$\mathcal{R}_m(\tilde{\mathcal{H}}) = \|\tilde{\mathbf{w}}\|_1\mathcal{R}_m(\tilde{\mathcal{F}}). \tag{37}$$

D.1. ResNet Module Hypothesis Space

Let n be the number of channels in ResNet, i.e., the number of input or output neurons in a module $\mathbf{f}_t(\mathbf{g}_t(x))$. We have proved that ResNet is equivalent as

$$F(x) = \mathbf{w}^\top\sum_{t=1}^{T}\mathbf{f}(\mathbf{g}_t(x)) \tag{38}$$

We define the family of functions that each neuron $f_{t,j}, \forall j \in [n]$ belong to as

$$\mathcal{F}_t = \{x \to \mathbf{u}_{t-1,j}(\sigma \circ \mathbf{f}_{t-1})(x) : \mathbf{u}_{t-1,j} \in \mathbb{R}^n, \|\mathbf{u}_{t-1,j}\|_1 \leq \Lambda_{t,t-1}, \mathbf{f}_{t-1,i} \in \mathcal{F}_{t-1}\} \tag{39}$$

where $\mathbf{u}_{t-1,j}$ denotes the vector of weights for connections from unit j to a lower layer $t-1$, $\sigma \circ \mathbf{f}_{t-1}$ denotes element-wise nonlinear transformation on $\mathbf{f}_{t-1}$. The output layer of each module is connected to the output layer of previous module. We consider 1-layer modules for convenience of analysis.

Therefore in ResNet with probability at least $1-\delta$,

$$\Pr_{\mathcal{D}}(yF(x) \leq 0) \leq \Pr_{S}(yF(x) \leq \theta) + \frac{4}{\theta}\sum_{t=1}^{T}\|\mathbf{w}\|_1 \mathcal{R}_m(\mathcal{F}_t) + \frac{2}{\theta}\sqrt{\frac{\log T}{m}} + \sqrt{\lceil \frac{4}{\theta^2}\log\left(\frac{\theta^2 m}{\log T}\right)\rceil \frac{\log T}{m} + \frac{\log\frac{2}{\delta}}{2m}} \tag{40}$$

for all $f_t \in \mathcal{F}_t$.

Define the maximum infinity norm over samples as $r_\infty \stackrel{\text{def}}{=} \mathbb{E}_{S\sim\mathcal{D}}\left[\max_{i\in[m]}\|x_i\|_\infty\right]$ and the product of l_1 norm bound on weights as $\Lambda_t \stackrel{\text{def}}{=} \prod_{t'=1}^{t} 2\Lambda_{t',t'-1}$. According to lemma 2 of (Cortes et al., 2016), the empirical Rademacher complexity is bounded as a function of r_∞, Λ_t and n:

$$\mathcal{R}_m(\mathcal{F}_t) \leq r_\infty \Lambda_t \sqrt{\frac{\log(2n)}{2m}} \tag{41}$$

Overall, with probability at least $1-\delta$,

$$\Pr_{\mathcal{D}}(yF(x) \leq 0) \leq \Pr_{S}(yF(x) \leq \theta) + \frac{4\|\mathbf{w}\|_1 r_\infty \sqrt{\frac{\log(2n)}{2m}}}{\theta}\sum_{t=1}^{T}\Lambda_t + \frac{2}{\theta}\sqrt{\frac{\log T}{m}} + \sqrt{\lceil \frac{4}{\theta^2}\log\left(\frac{\theta^2 m}{\log T}\right)\rceil \frac{\log T}{m} + \frac{\log\frac{2}{\delta}}{2m}} \tag{42}$$

for all $f_t \in \mathcal{F}_t$.

E. Proof for Theorem E: Margin and Generalization Bound

Theorem E.1. [***Generalization error bound***] *Given algorithm 1, the fraction of training examples with margin at most θ is at most $(1+\frac{2}{\frac{1}{\sqrt{\gamma_{T+1}}}-1})^{\frac{\theta}{2}}\exp(-\frac{1}{2}\gamma^2 T)$. And the generalization error $\Pr_D(yF(x)\leq 0)$ satisfies*

$$\Pr_{D}(yF(x) \leq 0) \leq (1+\frac{2}{\frac{1}{\gamma_{T+1}}-1})^{\frac{\theta}{2}}\exp(-\frac{1}{2}\gamma^2 T) + \frac{4C_0 r_\infty}{\theta}\sqrt{\frac{\log(2n)}{2m}}\sum_{t=1}^{T}\Lambda_t + \frac{2}{\theta}\sqrt{\frac{\log T}{m}} + \beta(\theta, m, T, \delta) \tag{43}$$

with probability at least $1-\delta$ for $\beta(\theta,m,T,\delta) \stackrel{\text{def}}{=} \sqrt{\left\lceil \frac{4}{\theta^2}\log\left(\frac{\theta^2 m}{\log T}\right)\right\rceil \frac{\log T}{m} + \frac{\log\frac{2}{\delta}}{2m}}$.

Now the proof for Theorem E is the following.

Proof. The fraction of examples in sample set S being smaller than θ is bounded

$$\Pr_S(yF(x) \le \theta) \le \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{y_i F(x_i) \le \theta\} \tag{44}$$

$$= \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{y_i \sum_{t=1}^{T} h_t(x_i) \le \theta\alpha_{T+1}\} \tag{45}$$

$$\le \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i \sum_{t=1}^{T} h_t(x_i) + \theta\alpha_{T+1}) \tag{46}$$

$$= \exp(\theta\alpha_{T+1})\frac{1}{m}\sum_{i=1}^{m} \exp(-y_i \sum_{t=1}^{T} h_t(x_i)) \tag{47}$$

$$= \exp(\theta\alpha_{T+1}) \prod_{t=1}^{T} Z_t \tag{48}$$

To bound $\exp(\theta\alpha_{T+1}) = \sqrt{(\frac{1+\tilde{\gamma}_{T+1}}{1-\tilde{\gamma}_{T+1}})^{\theta}}$, we first bound $\tilde{\gamma}_{T+1}$: We know that $\sum_{t=1}^{T}\prod_{t'=t+1}^{T}(1-\gamma_{t'}^2)\gamma_t^2 \le (1-\gamma^2)^{T-t}\gamma^2$ for all $\forall \gamma_t \ge \gamma^2 + \epsilon$ if $\gamma^2 \ge \frac{1-\epsilon}{2}$. Therefore $\forall\ \gamma_t \ge \gamma^2 + \epsilon$ and $\gamma^2 \ge \frac{1-\epsilon}{2}$

$$\tilde{\gamma}_{T+1}^2 = (1-\gamma_T^2)\tilde{\gamma}_T^2 + \gamma_T^2 \tag{49}$$

$$= \sum_{t=1}^{T}\prod_{t'=t+1}^{T}(1-\gamma_{t'}^2)\gamma_t^2 + \prod_{t=1}^{T}(1-\gamma_t^2)\tilde{\gamma}_1^2 \tag{50}$$

$$\le \sum_{t=1}^{T}(1-\gamma^2)^{T-t}\gamma^2 + (1-\gamma^2)^T\tilde{\gamma}_1^2 \tag{51}$$

$$= \sum_{t=0}^{T-1}(1-\gamma^2)^{t}\gamma^2 + (1-\gamma^2)^T\tilde{\gamma}_1^2 \tag{52}$$

$$= 1 - (1-\gamma^2)^T + (1-\gamma^2)^T\tilde{\gamma}_1^2 \tag{53}$$

$$= 1 - (1-\tilde{\gamma}_1^2)(1-\gamma^2)^T \tag{54}$$

Therefore

$$\Pr_{\mathcal{S}}(yF(x) \le \theta) \le \exp(\theta\alpha_{T+1}) \prod_{t=1}^{T} Z_t \tag{55}$$

$$= (\frac{1+\tilde{\gamma}_{T+1}}{1-\tilde{\gamma}_{T+1}})^{\frac{\theta}{2}} \prod_{t=1}^{T} Z_t \tag{56}$$

$$= (\frac{1+\tilde{\gamma}_{T+1}}{1-\tilde{\gamma}_{T+1}})^{\frac{\theta}{2}} \prod_{t=1}^{T} \sqrt{1-\gamma_t^2} \tag{57}$$

$$= (1 + \frac{2}{\frac{1}{\tilde{\gamma}_{T+1}} - 1})^{\frac{\theta}{2}} \exp(-\frac{1}{2}\gamma^2 T) \tag{58}$$

$$\le (1 + \frac{2}{\frac{1}{\sqrt{1-(1-\tilde{\gamma}_1^2)(1-\gamma^2)^T}} - 1})^{\frac{\theta}{2}} \exp(-\frac{1}{2}\gamma^2 T) \tag{59}$$

As $T \to \infty$, $\Pr_{\mathcal{S}}(yF(x) \le \theta) \le 0$ as $\exp(-\frac{1}{2}\gamma^2 T)$ decays faster than $(1 + \frac{2}{\frac{1}{\sqrt{1-(1-\tilde{\gamma}_1^2)(1-\gamma^2)^T}}-1})^{\frac{\theta}{2}}$. ∎

F. Telescoping Sum Boosting for Multi-calss Classification

Recall that the weak module classifier is defined as

$$h_t(x) = \alpha_{t+1} o_{t+1}(x) - \alpha_t o_t(x) \in \mathbb{R}^C, \tag{60}$$

where $o_t(x) \in \Delta^{C-1}$.

The weak learning condition for multi-class classification is different from the binary classification stated in the previous section, although minimal demands placed on the weak module classifier require prediction better than random on any distribution over the training set intuitively.

We now define the weak learning condition. It is again inspired by the slightly better than random idea, but requires a more sophisticated analysis in the multi-class setting.

F.1. Cost Matrix

In order to characterize the training error, we introduce the cost matrix $\mathbf{C} \in \mathbb{R}^{m \times C}$ where each row denote the cost incurred by classifying that example into one of the C categories. We will bound the training error using exponential loss, and under the exponential loss function defined as in Definition G.1, the optimal cost function used for best possible training error is therefore determined.

Lemma F.1. *The optimal cost function under the exponential loss is*

$$\mathbf{C}_t(i,l) = \begin{cases} \exp\left(s_t(x_i,l) - s_t(x_i,y_i)\right) & \text{if } l \neq y_i \\ -\sum\limits_{l' \neq y_i} \exp\left(s_t(x_i,l') - s_t(x_i,y_i)\right) & \text{if } l = y_i \end{cases} \tag{61}$$

where $s_t(x) = \sum\limits_{\tau=1}^{t} h_\tau(x)$.

F.2. Weak Learning Condition

Definition F.2. *Let* $\tilde{\gamma}_{t+1} = \frac{-\sum_{i=1}^{m} <\mathbf{C}_t(i,:), o_{t+1}(x_i)>}{\sum_{i=1}^{m}\sum_{l \neq y_i} \mathbf{C}_t(i,l)}$ *and* $\tilde{\gamma}_t = \frac{-\sum_{i=1}^{m} <\mathbf{C}_{t-1}(i,:), o_t(x_i)>}{\sum_{i=1}^{m}\sum_{l \neq y_i} \mathbf{C}_{t-1}(i,l)}$. *A multi-class weak module classifier* $h_t(x) = \alpha_{t+1} o_{t+1}(x) - \alpha_t o_t(x)$ *satisfies the* γ*-weak learning condition if* $\frac{\tilde{\gamma}_{t+1}^2 - \tilde{\gamma}_t^2}{1-\tilde{\gamma}_t^2} \geq \gamma^2 > 0$, *and* $Cov(< \mathbf{C}_t(i,:), o_{t+1}(x_i) >, < \mathbf{C}_t(i,:), o_{t+1}(x_i) >) \geq 0$.

We propose a novel learning algorithm using the optimal edge-over-random cost function for training ResNet under multi-class classification task as in Algorithm 3.

Theorem F.3. *The training error of a* T*-module ResNet using Algorithm 3 and 4 decays exponentially with the depth of the ResNet* T*,*

$$\frac{C-1}{m} \sum_{i=1}^{m} L_\eta^{exp}(s_T(x_i)) \leq (C-1) e^{-\frac{1}{2} T \gamma^2} \tag{62}$$

if the weak module classifier $h_t(x)$ *satisfies the* γ*-weak learning condition* $\forall t \in [T]$.

The exponential loss function defined as in Definition G.1

Algorithm 3 BoostResNet: telescoping sum boosting for multi-class classification

Input: Given $(x_1, y_1), \dots (x_m, y_m)$ where $y_i \in \mathcal{Y} = \{1, \dots, C\}$ and a threshold γ
Output: $\{f_t(\cdot), \forall t\}$ and W_{T+1} ▷ Discard $\mathbf{w}_{t+1}, \forall t \neq T$

Initialize $t \leftarrow 0$, $\tilde{\gamma}_0 \leftarrow 1$, $\alpha_0 \leftarrow 0$, $o_0 \leftarrow \mathbf{0} \in \mathbb{R}^C$, $s_0(x_i, l) = 0, \forall i \in [m], l \in \mathcal{Y}$
Initialize cost function $\mathbf{C}_0(i, l) \leftarrow \begin{cases} 1 & \text{if } l \neq y_i \\ 1 - C & \text{if } l = y_i \end{cases}$
while $\gamma_t > \gamma$ **do**
  $f_t(\cdot), \alpha_{t+1}, W_{t+1}, o_{t+1}(x) \leftarrow$ Algorithm 4$(g_t(x), \mathbf{C}_t, o_t(x), \alpha_t)$
  Compute $\gamma_t \leftarrow \sqrt{\frac{\tilde{\gamma}_{t+1}^2 - \tilde{\gamma}_t^2}{1 - \tilde{\gamma}_t^2}}$ ▷ where $\tilde{\gamma}_{t+1} \leftarrow \frac{-\sum_{i=1}^m \mathbf{C}_t(i,:) \cdot o_{t+1}(x_i)}{\sum_{i=1}^m \sum_{l \neq y_i} \mathbf{C}_t(i,l)}$
  Update $s_{t+1}(x_i, l) \leftarrow s_t(x_i, l) + h_t(x_i, l)$ ▷ where $h_t(x_i, l) = \alpha_{t+1} o_{t+1}(x_i, l) - \alpha_t o_t(x_i, l)$
  Update cost function $\mathbf{C}_{t+1}(i, l) \leftarrow \begin{cases} e^{s_{t+1}(x_i, l) - s_{t+1}(x_i, y_i)} & \text{if } l \neq y_i \\ -\sum_{l' \neq y_i} e^{s_{t+1}(x_i, l') - s_{t+1}(x_i, y_i)} & \text{if } l = y_i \end{cases}$
  $t \leftarrow t + 1$
end while
$T \leftarrow t - 1$

Algorithm 4 BoostResNet: oracle implementation for training a ResNet module (multi-class)

Input: $g_t(x)$,s_t,$o_t(x)$ and α_t
Output: $f_t(\cdot)$, α_{t+1}, W_{t+1} and $o_{t+1}(x)$

$(f_t, \alpha_{t+1}, W_{t+1}) \leftarrow \arg\min_{(f,\alpha,V)} \sum_{i=1}^m \sum_{l \neq y_i} e^{\alpha V^\top [f(g_t(x_i), l) - f(g_t(x_i), y_i) + g_t(x_i, l) - g_t(x_i, y_i)]}$
$o_{t+1}(x) \leftarrow W_{t+1}^\top [f_t(g_t(x)) + g_t(x)]$

F.3. Oracle Implementation

We implement an oracle to minimize $Z_t \overset{\text{def}}{=} \sum_{i=1}^m \sum_{l \neq y_i} e^{s_t(x_i, l) - s_t(x_i, y_i)} e^{h_t(x_i, l) - h_t(x_i, y_i)}$ given current state s_t and hypothesis module $o_t(x)$. Therefore minimizing Z_t is equivalent to the following.

$$\min_{(f,\alpha,V)} \sum_{i=1}^m \sum_{l \neq y_i} e^{s_t(x_i, l) - s_t(x_i, y_i)} e^{-\alpha_t (o_t(x_i, l) - o_t(x_i, y_i))} e^{\alpha V^\top [f(g_t(x_i), l) - f(g_t(x_i), y_i) + g_t(x_i, l) - g_t(x_i, y_i)]} \tag{63}$$

$$\equiv \min_{(f,\alpha,V)} \sum_{i=1}^m \sum_{l \neq y_i} e^{\alpha V^\top [f(g_t(x_i), l) - f(g_t(x_i), y_i) + g_t(x_i, l) - g_t(x_i, y_i)]} \tag{64}$$

$$\equiv \min_{\alpha, f, v} \sum_{i=1}^m e^{-\alpha v^\top [f(x_i, y_i) + g_t(x_i, y_i)]} \sum_{l \neq y_i} e^{\alpha v^\top [f(x_i, l) + g_t(x_i, l)]} \tag{65}$$

G. Proof for Theorem F.3 multiclass boosting theory

Proof. To characterize the training error, we use the exponential loss function

Definition G.1. *Define loss function for a multiclass hypothesis $H(x_i)$ on a sample (x_i, y_i) as*

$$L_\eta^{exp}(H(x_i), y_i) = \sum_{l \neq y_i} \exp\left((H(x_i, l) - H(x_i, y_i))\right). \tag{66}$$

Define the accumulated weak learner $s_t(x_i, l) = \sum_{t'=1}^t h_{t'}(x_i, l)$ and the loss $Z_t = \sum_{i=1}^m \sum_{l \neq y_i} \exp(s_t(x_i, l) - s_t(x_i, y_i)) \exp(h_t(x_i, l) - h_t(x_i, y_i))$.

Recall that $s_t(x_i, l) = \sum_{t'=1}^{t} h_{t'}(x_i, l) = \alpha_{t+1} W_{t+1}^\top g_{t+1}(x_i)$, the loss for a T-module multiclass ResNet is thus

$$\Pr_{i\sim D_1}(p(\alpha_{T+1}W_{T+1}^\top g_{T+1}(x_i)) \neq y_i) \leq \frac{1}{m}\sum_{i=1}^{m} L_\eta^{exp}(s_T(x_i)) \tag{67}$$

$$\leq \frac{1}{m}\sum_{i=1}^{m}\sum_{l\neq y_i} \exp\left(\eta(s_T(x_i,l) - s_T(x_i,y_i))\right) \tag{68}$$

$$\leq \frac{1}{m} Z_T \tag{69}$$

$$= \prod_{t=1}^{T} \frac{Z_t}{Z_{t-1}} \tag{70}$$

Note that $Z_0 = \frac{1}{m}$ as the initial accumulated weak learners $s_0(x_i, l) = 0$.

The loss fraction between module t and $t-1$, $\frac{Z_t}{Z_{t-1}}$, is related to $Z_t - Z_{t-1}$ as $\frac{Z_t}{Z_{t-1}} = \frac{Z_t - Z_{t-1}}{Z_{t-1}} + 1$.

The Z_t is bounded

$$Z_t = \sum_{i=1}^{m}\sum_{l\neq y_i} \exp(s_t(x_i,l) - s_t(i,y_i) + h_t(x_i,l) - h_t(x_i,y_i)) \tag{71}$$

$$\leq \sum_{i=1}^{m}\sum_{l\neq y_i} e^{s_t(x_i,l)-s_t(x_i,y_i)} e^{\alpha_{t+1}o_{t+1}(x_i,l) - \alpha_{t+1}o_{t+1}(x_i,y_i)} \sum_{i=1}^{m}\sum_{l\neq y_i} e^{s_t(x_i,l)-s_t(x_i,y_i)} e^{-\alpha_t o_t(x_i,l) + \alpha_t o_t(x_i,y_i)} \tag{72}$$

$$\leq \sum_{i=1}^{m}\sum_{l\neq y_i} e^{s_t(x_i,l)-s_t(x_i,y_i)} \left(\frac{e^{-\alpha_{t+1}} + e^{\alpha_{t+1}}}{2} + \frac{e^{-\alpha_{t+1}} - e^{\alpha_{t+1}}}{2}\left(o_{t+1}(x_i,y_i) - o_{t+1}(x_i,l)\right)\right)$$
$$\sum_{i=1}^{m}\sum_{l\neq y_i} e^{s_{t-1}(x_i,l)-s_{t-1}(x_i,y_i)} \left(\frac{e^{\alpha_t} + e^{-\alpha_t}}{2}\right) \tag{73}$$

$$= \left(\frac{e^{-\alpha_{t+1}} + e^{\alpha_{t+1}} - 2}{2} Z_{t-1} + \frac{e^{\alpha_{t+1}} - e^{-\alpha_{t+1}}}{2} \sum_{i=1}^{m} < C_t(x_i,:), o_{t+1}(x_i,:) > \right)\left(\frac{e^{\alpha_t} + e^{-\alpha_t}}{2}\right)$$

$$\leq \left(\frac{e^{-\alpha_{t+1}} + e^{\alpha_{t+1}} - 2}{2} Z_{t-1} + \frac{e^{\alpha_{t+1}} - e^{-\alpha_{t+1}}}{2} \sum_{i=1}^{m} < C_t(x_i,:), U_{\tilde{\gamma}_t}(x_i,:) > \right)\left(\frac{e^{\alpha_t} + e^{-\alpha_t}}{2}\right) \tag{74}$$

$$= \left(\frac{e^{-\alpha_{t+1}} + e^{\alpha_{t+1}} - 2}{2} Z_{t-1} + \frac{e^{\alpha_{t+1}} - e^{-\alpha_{t+1}}}{2} (-\tilde{\gamma}_t) Z_{t-1}\right)\left(\frac{e^{\alpha_t} + e^{-\alpha_t}}{2}\right) \tag{75}$$

Therefore

$$\frac{Z_t}{Z_{t-1}} \leq \left(\frac{e^{-\alpha_{t+1}} + e^{\alpha_{t+1}}}{2} + \frac{e^{-\alpha_{t+1}} - e^{\alpha_{t+1}}}{2}\tilde{\gamma}_t\right)\left(\frac{e^{\alpha_t} + e^{-\alpha_t}}{2}\right) \tag{76}$$

The algorithm chooses α_{t+1} to minimize Z_t. We achieve an upper bound on Z_t, $\sqrt{\frac{1-\tilde{\gamma}_t^2}{1-\tilde{\gamma}_{t-1}{}^2}}$

by minimizing the bound in Equation (76)

$$Z_t|_{\alpha_{t+1}=\arg\min Z_t} \leq Z_t|_{\alpha_{t+1}=\frac{1}{2}\ln\left(\frac{1+\tilde{\gamma}_t}{1-\tilde{\gamma}_t}\right)} \tag{77}$$

$$\leq \left(\frac{e^{-\alpha_{t+1}} + e^{\alpha_{t+1}}}{2} + \frac{e^{-\alpha_{t+1}} - e^{\alpha_{t+1}}}{2}\tilde{\gamma}_t\right)\frac{e^{\alpha_t} + e^{-\alpha_t}}{2}\Bigg|_{\alpha_{t+1}=\frac{1}{2}\ln\left(\frac{1+\tilde{\gamma}_t}{1-\tilde{\gamma}_t}\right)} \tag{78}$$

$$= \sqrt{\frac{1-\tilde{\gamma}_t^2}{1-\tilde{\gamma}_{t-1}^2}} = \sqrt{1-\gamma_t^2} \tag{79}$$

Therefore over the T modules, the training error is upper bounded as follows

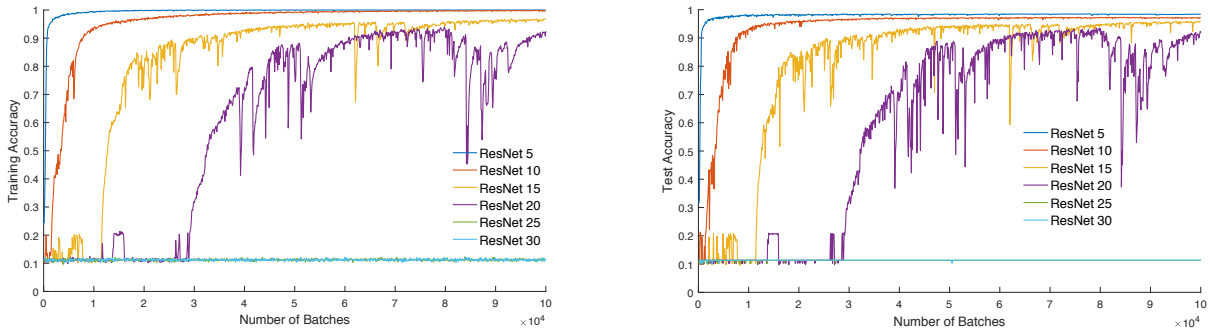
$$\Pr_{i\sim D}(p(\alpha_{T+1}w_{T+1}^{\top}g_{T+1}(x_i))) \neq y_i) \leq \prod_{t=1}^{T}\sqrt{1-\gamma_t^2} \leq \prod_{t=1}^{T}\sqrt{1-\gamma^2} = \exp\left(-\frac{1}{2}T\gamma^2\right) \tag{80}$$

Overall, Algorithm 3 and 4 leads us to consistent learning of ResNet. □

H. Experiments

H.1. Training error degradation of e2eBP on ResNet

We investigate e2eBP training performance on various depth ResNet. Surprisingly, we observe a training error degradation for *e2eBP* although the ResNet's identity loop is supposed to alleviate this problem. Despite the presence of identity loops, the *e2eBP* eventually is susceptible to spurious local optima. This phenomenon is explored further in Figures 5a and 5b, which respectively show how training and test accuracies vary throughout the fitting process. Our proposed sequential training procedure, *BoostResNet*, relieves gradient instability issues, and continues to perform well as depth increases.



(a) e2eBP training accuracy

(b) e2eBP test accuracy

Figure 5: Convergence of *e2eBP* (baseline) on multilayer perceptron residual network (of various depths) on MNIST dataset.