SPURIOUS LOCAL MINIMA PROVABLY EXIST FOR DEEP CONVOLUTIONAL NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we prove that a general family of spurious local minima exist in the loss landscape of deep convolutional neural networks with squared loss or cross-entropy loss. For this purpose, we develop some new techniques to solve the challenges introduced by convolutional layers. We solve a combinatorial problem which considers the limited receptive fields of hidden neurons, and possible distinct activation status for different samples and different locations in feature maps, to show that a differentiation of data samples is always possible somewhere in feature maps. Training loss is then decreased by perturbation of network parameters that can affect different samples in different ways. Despite filters and biases are tied in each feature map, we give a construction in which this perturbation only affects the output of a single ReLU neuron and keeps the outputs at other locations unchanged. Finally, we give an example of nontrivial spurious local minimum in which different activation patterns of samples are explicitly constructed. Experimental results verify our theoretical findings.

1 INTRODUCTION

Convolutional neural networks (CNNs) (e.g. Lecun et al. (1998); Krizhevsky et al. (2012); Simonyan & Zisserman (2015); Szegedy et al. (2015); He et al. (2016); Huang et al. (2017)), one of the most important models in deep learning, have been successfully applied to many domains. Spurious local minima, whose losses are greater than that of global minimum, play an important role in the training of deep CNNs and understanding of deep learning models. It is widely believed that spurious local minima exist in the loss landscape of CNNs which is thought to be highly non-convex, as evidenced by some experimental studies (e.g. Dauphin et al. (2014); Goodfellow et al. (2015); Liao & Poggio (2017); Freeman & Bruna (2017); Draxler et al. (2018); Garipov et al. (2018); Li et al. (2018); Mehmeti-Gopel et al. (2021)). However, the existence of spurious local minima for CNNs caused by convolutions has never been proved mathematically before.

Showing existence of spurious local minima in the loss landscape of CNNs faces some challenges. For fully connected deep ReLU networks, previous works (He et al. (2020); Ding et al. (2019); Goldblum et al. (2020); Liu et al. (2021)) usually split data samples into some groups according to the inputs of a specific ReLU neuron, such that each group will have different activation patterns and thus behave differently under the perturbation of network parameters and then a lower loss can be resulted. This technique relies on data split and parameter perturbation, and cannot be directly applied to CNNs due to the following difficulties. Every neuron in CNN feature maps has limited receptive field that covers only partial pixels in an input image (take image as an example), and hence the inputs to a ReLU neuron can be identical even for distinct samples, making them hard to distinguish. This data split issue is further complicated by the nonlinear ReLU activations that truncate negative inputs, and the activation status can vary from place to place and from sample to sample. Moreover, the filters and biases of CNNs are shared by all neurons in the same feature map, and thus adjusting the output of a ReLU neuron by perturbating these tied parameters will also affect other neurons in the same feature map.

We solve these challenges by developing some new techniques in this paper. By taking limited receptive fields and possible distinct activation status for different locations and samples into account, we solve a combinatorial problem to show that a split of data samples is always possible somewhere in feature maps. We then present a construction of CNN parameters that can be perturbated to achieve

lower losses for some general local minima. Despite the parameters are tied, our construction can perturbate the outputs of samples at a single neuron in feature maps without affecting other locations. We also give a concrete example of spurious local minima for CNNs. To our best knowledge, this is the first work showing existence of spurious local minima in CNNs introduced by convolutional layers.

This paper is organized as follows. Section 1.1 is related work. Section 2 describes convolutional neural networks, and gives some notations used in this paper. In section 3, our general results on spurious local minima are given. In section 4, we present an example of nontrivial spurious local minima for CNNs. Section 5 presents experimental results to verify our theoretical findings. Finally, conclusions and future directions are provided. More auxiliary lemmas, experimental details and all proofs are given in appendices.

1.1 Related Work

For some neural networks and learning models, it has been shown that there exist no spurious local minima. These models include deep linear networks (Baldi & Hornik (1989); Kawaguchi (2016); Lu & Kawaguchi (2017); Laurent & von Brecht (2018); Yun et al. (2018); Nouiehed & Razaviyayn (2018); Zhang (2019)), matrix completion and tensor decomposition (e.g., Ge et al. (2016)), one-hidden-layer networks with quadratic activation (Soltanolkotabi et al. (2019); Du & Lee (2018)), deep linear residual networks (Hardt & Ma (2017)) and deep quadratic networks (Kazemipour et al. (2020)).

Existence of spurious local minima for one-hidden-layer ReLU networks has been demonstrated, by constructing examples of networks and data samples, in Safran & Shamir (2018); Swirszcz et al. (2016); Zhou & Liang (2018); Yun et al. (2019); Ding et al. (2019); Sharifnassab et al. (2020); He et al. (2020); Goldblum et al. (2020) etc.

For deep ReLU networks, He et al. (2020); Ding et al. (2019); Goldblum et al. (2020); Liu et al. (2021) showed that spurious local minima exist for fully connected deep neural networks with some general loss functions. For these spurious local minima, all ReLU neurons are active and deep neural networks are reduced to linear predictors. Spurious local minima for CNNs are not treated in these works. In comparison, we deal with spurious local minima for CNNs in this work, and the constructed spurious local minima can be nontrivial in which nonlinear predictors are generated and some ReLU neurons are inactive.

Du et al. (2018) showed the existence of spurious local minima for one-hidden-layer CNNs with a single non-overlapping filter, Gaussian input and squared loss. In contrast, we discuss practical deep CNNs with multiple filters of overlapping receptive fields and arbitrary input, for both squared and cross-entropy loss. Nguyen & Hein (2018) showed that a sufficiently wide CNN (include a wide layer which has more neurons than the number of training samples followed by a fully connected layer) has a well-behaved loss surface with almost no bad local minima. Liu (2022) explored spurious local minima for CNNs introduced by fully connected layers.

2 Preliminaries

2.1 Notations

We use $M_{i,\cdot}$ and $M(i,\cdot)$ to denote the ith row of a matrix M, and use $M_{\cdot,i}$ and $M(\cdot,i)$ to denote the ith column of M. $M_{i,j}$ and M_{ij} are the (i,j) entry of M. The ith component of a vector $\mathbf{v}$ is denoted as v_i, v^i or $v(i)$. $[N]$ is equivalent to $\{1,2,\cdots,N\}$. $i:j$ means $i,i+1,\cdots,j$. $v(i:end)$ stands for the components of a vector $\mathbf{v}$ from the ith one to the last one. $\mathbf{1}_n$ denotes a vector of size n whose components are all 1s.

2.2 Convolutional Neural Networks

A typical CNN includes some convolutional layers, pooling layers and fully-connected layers.

Convolutional layers can take advantage of the translational invariance inherent in data, and are defined as follows. Suppose the lth layer is a convolutional layer, neighboring neurons at layer $(l-1)$

are grouped into patches for the convolution operation. Let P_l and s_l be, respectively, the number of patches and the size of each patch in the lth layer. Denote $\left\{\mathbf{o}_1^l, \cdots, \mathbf{o}_{P_l}^l\right\} \in \mathbb{R}^{s_l}$ as the set of patches in the lth layer, and denote by T_l the number of convolutional filters in the lth layer. The number of neurons in the lth layer is then $n_l = T_l P_{l-1}$. Given the pth ($p \in [P_{l-1}]$) patch $\mathbf{o}_p^{l-1}$ and the tth ($t \in [T_l]$) filter $\mathbf{w}_t^l$, the output of corresponding neuron in layer l is given as follows,

$$o^l(h) = \sigma(\mathbf{w}_t^l \cdot \mathbf{o}_p^{l-1} + b_t^l), \tag{1}$$

where $h = (t-1)P_{l-1} + p$. $\sigma(x) = max(0, x)$ is the ReLU activation function. There is a single bias b_t^l for each feature map.

The convolution operation can be reformulated as a matrix-vector product. For example, given an input $\mathbf{o}^{l-1} = (a, b, c, d, e)^T$ and two filters $\mathbf{w}_1 = (w_1^1, w_1^2, w_1^3)^T$, $\mathbf{w}_2 = (w_2^1, w_2^2, w_2^3)^T$, the convolution operation is equivalent to

$$\mathbf{o}^l = \sigma\left[\begin{pmatrix} w_1^1 & w_1^2 & w_1^3 & 0 & 0 \\ 0 & w_1^1 & w_1^2 & w_1^3 & 0 \\ 0 & 0 & w_1^1 & w_1^2 & w_1^3 \\ w_2^1 & w_2^2 & w_2^3 & 0 & 0 \\ 0 & w_2^1 & w_2^2 & w_2^3 & 0 \\ 0 & 0 & w_2^1 & w_2^2 & w_2^3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} + \begin{pmatrix} b_1^l \\ b_1^l \\ b_1^l \\ b_2^l \\ b_2^l \\ b_2^l \end{pmatrix}\right]. \tag{2}$$

We will use matrix-vector product to represent convolution operations. Denote the weight matrix and bias vector of the lth layer as W^l and $\mathbf{b}^l$, respectively, with $W^l \in \mathbb{R}^{n_l \times n_{l-1}}$ and $\mathbf{b}^l \in \mathbb{R}^{n_l}$, then the output of the lth layer is $\mathbf{o}^l = \sigma(W^l \mathbf{o}^{l-1} + \mathbf{b}^l)$. This matrix-vector product formulation can also be used to represent fully connected layers.

In a max-pooling layer, the output of the pth neuron $o^l(p)$ is computed by

$$o^l(p) = \max(o_p^{l-1}(1), \cdots, o_p^{l-1}(s_{l-1})),\ p \in [P_{l-1}], \tag{3}$$

where $o_p^{l-1}(i)$ is the ith element in the pth patch of layer $(l-1)$. Similarly, a layer is called average pooling layer if the average of every patch is computed.

Consider a training set $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^{d_x}, \mathbf{y}_i \in \mathbb{R}^{d_y}$ $(i \in [N])$ are, respectively, the input and target output. Let L be the number of layers in a CNN, and let $\mathbf{o}^{l,i}$ and $\mathbf{o}^l(\mathbf{x}_i)$ be the output $\mathbf{o}^l$ for the ith data sample (denote $\mathbf{o}^{0,i} := \mathbf{x}_i$). The output vector of a CNN is

$$\mathbf{o}^{L,i} = W^L \mathbf{o}^{L-1,i} + \mathbf{b}^L,\ \ i \in [N].$$

We also use $\mathbf{o}_i$, $\mathbf{o}(\mathbf{x}_i)$ and $\mathbf{o}^L(\mathbf{x}_i)$ to denote $\mathbf{o}^{L,i}$.

We introduce a diagonal matrix $I^{l,i} \in \mathbb{R}^{n_l \times n_l}$ for each sample and each ReLU layer to represent the activation status of ReLU neurons, whose diagonal entries are $I_{k,k}^{l,i} = 1$ if $W^l(k,\cdot)\mathbf{o}^{l-1,i} + b_k^l > 0$ and $I_{k,k}^{L,i} = 0$ otherwise. Consequently, the output of a convolutional layer can be written as $\mathbf{o}^{l,i} = I^{l,i}(W^l \mathbf{o}^{l-1,i} + \mathbf{b}^l)$. The CNN output can be written as

$$\mathbf{o}_i = W^L I^{L-1,i}\left(W^{L-1} \cdots \left(W^1 \mathbf{x}_i + \mathbf{b}^1\right) + \cdots + \mathbf{b}^{L-1}\right) + \mathbf{b}^L. \tag{4}$$

The empirical risk (training loss) is defined as

$$R(W^1, \mathbf{b}^1, \cdots, W^L, \mathbf{b}^L) = \frac{1}{N}\sum_{i=1}^{N} l\left(\mathbf{o}_i, \mathbf{y}_i\right), \tag{5}$$

where l is the loss function, such as the widely used cross-entropy loss for classification problems. Our main goal in this paper is to construct spurious local minima for the empirical risk function of deep CNNs given in (5).

For ease of presentation, we will assume that the strides for convolutions are equal to one and the input feature maps are padded such that for each convolutional layer, the sizes of input and output feature maps are equal. The average pooling operation can be represented by a matrix-vector product. Within a small neighborhood around $\left(W^l, \mathbf{b}^l\right)_{l=1}^L$ in parameter space that keeps

$i_p^* := \arg\max_{i\in[s_{l-1}]}(o_p^{l-1}(i)),\ \forall p \in [P_{l-1}]$ constant, the max-pooling operation can be represented by a matrix-vector product as well. Therefore, we also use parameter matrix W^l to represent a pooling layer. Finally, because spurious local minima caused by fully connected layers in CNNs can be treated in the same way as in fully connected networks, we will focus on spurious local minima caused by convolutional layers.

Sometimes the weight matrices are augmented to include biases, and the output vector of each layer is augmented with a scalar 1. The output of the lth layer is then written as $\hat{\mathbf{o}}^{l,i} = \hat{I}^{l,i}\left(\hat{W}^l\hat{\mathbf{o}}^{l-1,i}\right),\ 1 \leq l \leq L-1$, where $\hat{W}^l$ and $\hat{I}^{l,i}$ are the augmented matrices, and $\hat{\mathbf{o}}^{l,i}$ is the augmented output vector.

3 Spurious Local Minima

In this section, we first construct a general local minimum with parameters $\theta := \left(W^l, \mathbf{b}^l\right)_{l=1}^{L}$. Then, we present another point θ' in parameter space which has an equal empirical risk with θ. We further perturbate θ' and show that empirical risk can be decreased. Therefore, θ is a spurious local minimum in parameter space.

3.1 Main Results

We first give a general construction of nontrivial local minima. Given a local minimum for a subnetwork of a CNN, we show that it will induce a local minimum of the whole CNN if the subnetwork is embedded into the CNN appropriately. Local minima for subnetworks of a CNN can be constructed using any method. Our construction of spurious local minima is general and does not rely on the concrete form of local minima for subnetworks. In section 4, we will give an explicit construction of exemplar nontrivial local minima for subnetworks.

For ease of presentation, we will assume that the last three hidden layers before the output layer (the Lth layer, which is fully connected) are all convolutional layers. We will fix the parameters of the last two convolutional layers in a way such that the output $\mathbf{o}^{L-3}(\mathbf{x})$ is passed through layers $L-2$ and $L-1$ without changing it. Two feature maps in layers $L-2$ and $L-1$, respectively, are reserved for the perturbation described later. Without loss of generality, let them be the first and second feature maps in these two layers, respectively. Note that the size of output feature map of the lth layer is P_{l-1}, and $P_{L-2} = P_{L-3} = P_{L-4}$ after padding.

Lemma 1. *Given a CNN whose numbers of filters in the last two convolutional layers satisfy $T_{L-1} \geq 3,\ T_{L-2} \geq 3$, the parameters of its last three layers are set as follows to propagate $\mathbf{o}^{L-3}(\mathbf{x})$ to the output layer. The first and second feature maps in both layer $L-2$ and layer $L-1$ have no contributions to the final output.*

$$\begin{aligned}
&W^L\left(\cdot, 1:2P_{L-2}\right) = 0,\\
&W^{L-1}\left(i,i\right) = 1,\ i \in [\mathit{min}\left(n_{L-1}, n_{L-2}\right)]\,; \quad W^{L-1}\left(i,j\right) = 0,\ i \neq j,\\
&W^{L-2}\left(i,\cdot\right) = 0,\ i \in [2P_{L-3}]\,; \quad W^{L-2}\left(2P_{L-3}+i, i\right) = 1,\ i \in [\mathit{min}\left(n_{L-2}-2P_{L-3}, n_{L-3}\right)]\,,\\
&W^{L-2}\left(2P_{L-3}+i, j\right) = 0,\ i \neq j,\ i \in [\mathit{min}\left(n_{L-2}-2P_{L-3}, n_{L-3}\right)]\,,\\
&\mathbf{b}^{L-1} = \mathbf{0},\ \mathbf{b}^{L-2} = \mathbf{0}.
\end{aligned} \tag{6}$$

The remaining parameters, including $W^L(\cdot, 2P_{L-2}+1:end),\ \mathbf{b}^L$ and $\left(W^l, \mathbf{b}^l\right)_{l=1}^{L-3}$, constitute a subnetwork, and they locally minimize the following training loss when fixing those parameters in (6),

$$\begin{aligned}
&\tilde{R}(\left(W^l, \mathbf{b}^l\right)_{l=1}^{L-3}, W^L(\cdot, 2P_{L-2}+1:end),\ \mathbf{b}^L)\\
&:= \frac{1}{N}\sum_{i=1}^{N} l\left(W^L I^{L-1,i}\left(W^{L-1}\cdots\left(W^1\mathbf{x}_i + \mathbf{b}^1\right) + \cdots + \mathbf{b}^{L-1}\right) + \mathbf{b}^L, \mathbf{y}_i\right).
\end{aligned} \tag{7}$$

Assume that the matrix $\hat{W}^L\hat{I}^{L-1,i}\hat{W}^{L-1}\cdots\hat{W}^1$ is full rank in a small neighborhood around $(\left(W^l, \mathbf{b}^l\right)_{l=1}^{L-3}, W^L(\cdot, 2P_{L-2}+1:end),\ \mathbf{b}^L)$ for all $i \in [N]$. Then, the point $\theta := \left(W^l, \mathbf{b}^l\right)_{l=1}^{L}$ obtained above is a local minimum in parameter space.

The full rank requirement in Lemma 1 is used to prevent degenerate cases, e.g. all ReLU neurons in some layers are deactivated for a sample.

If there exist pooling operations at the top layers of a CNN, the conclusion in Lemma 1 still holds. We only need to consider the top three convolutional layers and modify their parameters to reflect the effects of pooling operations.

The embedding scheme to construct local minima or saddle points has also been used for fully connected networks by Fukumizu et al. (2019); Fukumizu & Amari (2000); He et al. (2020); Liu et al. (2021); Zhang et al. (2022). However, embedding for CNNs needs to tackle the problems of limited receptive fields, arbitrary activation status at different locations, and tied weights and biases.

Our construction of spurious local minima relies on the following assumption.

Assumption 1. At local minimum θ, the inputs to the final fully connected layer are distinct for different samples, i.e., $\forall i, j \in [N]$ and $i \neq j,\ \mathbf{o}^{L-1,i} \neq \mathbf{o}^{L-1,j}$.

For practical CNNs and datasets with rich contents, the number of neurons n_{L-1} input to the final fully connected layer is big and the chance of $\mathbf{o}^{L-1,i}$ (note that $\mathbf{o}^{L-1,i} \neq \mathbf{0}$ since $I^{L-1,i} \neq 0$ by the full rank requirement in Lemma 1) being identical for different samples is very small, thus Assumption 1 is reasonable and enforces a very mild restriction on local minimum θ. This assumption is used to prevent distinct samples from becoming indistinguishable when passing through ReLU layers and being truncated by them.

In the following, we will show that given Assumption 1, there must exist at least one location in feature maps where the outputs of all samples can be split into two parts using a threshold such that the two parts will behave differently under perturbations. This is not a easy task due to the following difficulties introduced by convolutional layers. Firstly, the receptive field of each neuron in a convolutional layer is smaller than the input image, thus at each location in feature maps, the effective inputs (pixels in the receptive field) from distinct samples can be identical and indistinguishable. Secondly, the activation status of hidden ReLU neurons can vary from place to place and from sample to sample. Assumption 1 and the full rank assumption in Lemma 1 will help us to distinguish different samples. Thirdly, the filters and biases are shared by all locations in a feature map, and one cannot perturbate these parameters at a location without affecting other places.

We assume that the training loss at θ satisfies $R(\left(W^l, \mathbf{b}^l\right)_{l=1}^L) > 0$. By Lemma 6 in Appendix B, for squared and cross-entropy loss function l, $\frac{\partial l(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i)}{\partial \mathbf{o}}$ cannot equal to zero for all samples if $R(\theta) > 0$. Without loss of generality, assume $u_i := \frac{\partial l(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i)}{\partial o(1)} \neq 0$ for some $i \in [N]$, where $o(1)$ is the first component of output vector $\mathbf{o}$. By optimality of $\theta := \left(W^l, \mathbf{b}^l\right)_{l=1}^L$, we have $\frac{\partial R}{\partial b_1^L} = \frac{1}{N} \sum_{i=1}^N \frac{\partial l(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i)}{\partial o(1)} = \frac{1}{N} \sum_{i=1}^N u_i = 0$.

We will split the outputs of all samples at layer $L-3$. Given CNN parameters $\theta := \left(W^l, \mathbf{b}^l\right)_{l=1}^L$ and data samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, we consider the outputs $o^{L-3,i}(j)$ $(i \in [N], j \in [\min(n_{L-1} - 2P_{L-2}, n_{L-2} - 2P_{L-3}, n_{L-3})])$ in feature maps of layer $L-3$. For notational simplicity, we denote $M := \min(n_{L-1} - 2P_{L-2}, n_{L-2} - 2P_{L-3}, n_{L-3})$, and for all $j \in [M],\ i \in [N]$

$$
\begin{aligned}
&v_i^j := W^{L-3}(j, \cdot)\, \mathbf{o}^{L-4,i}, \qquad w^j := W^L(1, 2P_{L-2} + j), \\
&o_i := o^{L,i}(1) = \sum_{j=1}^{M} w^j \sigma\left(v_i^j + b_{t_j}^{L-3}\right) + b_1^L.
\end{aligned}
\tag{8}
$$

where $b_{t_j}^{L-3}$ is the bias associated with the feature map in which location j resides. There are in total M locations in the feature maps of layer $L-3$. v_i^j is the output at layer $L-3$ before adding bias and ReLU activation. Thus, at each location $j \in [M]$, there is a list $(v_1^j, v_2^j, \cdots, v_N^j)$. Let $\tilde{\mathbf{x}}_i^j$ be the vector composed of pixels of $\mathbf{x}_i$ in the receptive field of the jth location. $\tilde{\mathbf{x}}_i^j$ can be identical for distinct samples, resulting in identical v_i^js. Identical v_i^js can also be resulted from ReLU activations that remove the difference between input samples.

In general, at each location j in the feature maps, $(v_1^j, v_2^j, \cdots, v_N^j)$ are organized in groups according to their magnitudes. Assume there are g_j groups in the list $(v_1^j, v_2^j, \cdots, v_N^j)$. Without loss of

generality, we can re-index the samples in descending order of v_i^j and write the ordered list as $v_1^j = v_2^j = \cdots v_{m_1}^j > v_{m_1+1}^j = \cdots v_{m_2}^j > \cdots > v_{m_{g_j-1}+1}^j = \cdots = v_{m_{g_j}}^j = v_N^j$.

Let $\mathbf{u} = (u_1, u_2, \cdots, u_N)^T$ and note that $\mathbf{u} \neq \mathbf{0}$ and $\sum_{i=1}^N u_i = 0$. In the following lemma, we show that there is at least one location where the split of data samples exists. This is a combinatorial problem which needs to deal with multiple locations and multiple ordered groups at each location, and the groupings of samples can be arbitrary at each location.

Lemma 2. *Assume that training loss $R(\left(W^l, \mathbf{b}^l\right)_{l=1}^L) > 0$ for squared or cross-entropy loss. Under Assumption 1, there exist some locations $j \in [M]$ where the ordered list $v_1^j = v_2^j = \cdots v_{m_1}^j > v_{m_1+1}^j = \cdots v_{m_2}^j > \cdots > v_{m_{g_j-1}+1}^j = \cdots = v_{m_{g_j}}^j = v_N^j$ can be splitted into two parts $\left\{v_1^j, v_2^j, \cdots, v_n^j\right\}$ and $\left\{v_{n+1}^j, \cdots, v_N^j\right\}$ by a threshold η such that*

$$v_1^j, v_2^j, \cdots, v_n^j > \eta, \;\; v_{n+1}^j, v_{n+2}^j, \cdots, v_N^j < \eta, \;\; \sum_{i=1}^n u_i \neq 0. \tag{9}$$

There may be more than one location where such data split exists, and we choose the one with the biggest threshold η for later usage. Let h be the location with the biggest split threshold, and without loss of generality, assume it is located in the first feature map of layer $L-3$ (that is why we put a special emphasis on the first and second feature maps in layers $L-2$ and $L-1$ in Lemma 1 and Lemma 3). Then, we have

$$\forall j \in [M], \; j \neq h, \sum_{i \in \left\{1,2,\cdots,N \mid v_i^j > \eta\right\}} u_i = 0. \tag{10}$$

Given the CNN parameters $\theta := (W^l, \mathbf{b}^l)_{l=1}^L$ specified in Lemma 1 and the threshold η in Lemma 2, we give the following point $\theta' := \left(W'^l, \mathbf{b}'^l\right)_{l=1}^L$ in parameter space and show that training loss at θ' is equal to that at θ. We only change the parameters in (6) that are related with the first and second feature maps in layers $L-2$ and $L-1$, respectively, to obtain θ', and remaining parameters are fixed. The purpose of the following parameter setting is to enable different behaviors under perturbation of parameters for different parts of samples. Different settings are designed depending on whether $\eta \geq -b_1^{L-3}$ or not.

Lemma 3. *Given the CNN parameters $\theta := (W^l, \mathbf{b}^l)_{l=1}^L$ specified in Lemma 1, let $\theta' := \left(W'^l, \mathbf{b}'^l\right)_{l=1}^L$. If $\eta \geq -b_1^{L-3}$, set*

$$\begin{aligned}
&W'^L\left(1, P_{L-2}+1 : 2P_{L-2}\right) = W^L\left(1, 2P_{L-2}+1 : 3P_{L-2}\right), \\
&W'^L\left(1, 1 : P_{L-2}\right) = W^L\left(1, 2P_{L-2}+1 : 3P_{L-2}\right), \quad W^L\left(1, 2P_{L-2}+1 : 3P_{L-2}\right) = \mathbf{0}^T, \\
&W'^{L-1}(i,i) = -1, \; i \in [P_{L-3}]; \qquad W'^{L-1}(i,i) = +1, \; i \in \{P_{L-3}+1 : 2P_{L-3}\}, \\
&b_1'^{L-1} = \eta + b_1^{L-3}, \qquad b_2'^{L-1} = \eta + b_1^{L-3}, \\
&W'^{L-2}(i,i) = -1, \; i \in [P_{L-4}]; \qquad W'^{L-2}(P_{L-3}+i, i) = +1, \; i \in [P_{L-4}], \\
&b_1'^{L-3} = -\min_{i \in [N], \; j \in [P_{L-4}]} v_i^j, \qquad b_1'^{L-2} = \eta + b_1'^{L-3}; \\
&b_2'^{L-2} = -\eta - b_1'^{L-3}; \qquad b_1'^L = b_1^L - \sum_{j=1}^{P_{L-2}} w^j \left(\eta + b_1^{L-3}\right).
\end{aligned} \tag{11}$$

If $\eta < -b_1^{L-3}$*, set*

$$W'^L(i, 1:P_{L-2}) = W^L(i, 2P_{L-2}+1:3P_{L-2}),$$
$$W'^L(i, P_{L-2}+1:2P_{L-2}) = -W^L(i, 2P_{L-2}+1:3P_{L-2}),\ i \in [d_y];$$
$$W'^{L-1}(i,i) = 1,\ i \in [P_{L-3}]; \qquad W'^{L-1}(i,i) = 1,\ i \in [P_{L-3}+1:2P_{L-3}];$$
$$W'^{L-1}(i,i) = 1,\ i \in [2P_{L-3}+1:3P_{L-3}],$$
$$W'^{L-2}(i,i) = 1,\ i \in [P_{L-4}]; \qquad W'^{L-2}(P_{L-3}+i, i) = 1,\ i \in [P_{L-4}],$$
$$b_1'^{L-3} = -\min_{i\in[N],\ j\in[P_{L-4}]} v_i^j, \qquad b_1'^{L-2} = -\eta - b_1'^{L-3}, \qquad b_2'^{L-2} = -\eta - b_1'^{L-3},$$
$$b_3'^{L-2} = -\eta - b_1'^{L-3}, \qquad b_1'^{L-1} = 0, \qquad b_2'^{L-1} = 0, \qquad b_3'^{L-1} = \eta + b_1^{L-3}; \qquad b_i'^L = b_i^L,\ i \in [d_y]. \tag{12}$$

Remaining parameters in θ *keep fixed. Then,* $R(\theta') = R(\theta)$*.*

The purpose of introducing $b_1'^{L-3}$ is to make $\sigma(v_i^j + b_1'^{L-3}) \geq 0$ for all $i \in [N],\ j \in [P_{L-4}]$.

Next, we will show that by perturbating θ', we can decrease the training loss. We use different perturbation schemes for different cases of η.

Lemma 4. *If* $\eta \geq -b_1^{L-3}$*, perturbate* $b_2'^{L-2} = -\eta - b_1'^{L-3}$ *as*

$$b_2'^{L-2} \to b_2'^{L-2} + \delta b = -\eta - b_1'^{L-3} + \delta b. \tag{13}$$

If $\eta < -b_1^{L-3}$*, perturbate* $b_1'^{L-2} = -\eta - b_1'^{L-3}$ *as*

$$b_1'^{L-2} \longrightarrow b_1'^{L-2} + \delta b = -\eta - b_1'^{L-3} + \delta b. \tag{14}$$

Remaining parameters in θ' *keep fixed. Then under Assumption 1, a training loss lower than* $R(\theta')$ *is obtained under this perturbation with a proper sign of* δb*.*

Only the output at location h will be perturbated under the perturbation in Lemma 4 despite the tied biases. The detail is given in the proof in section D.4.

Combining Lemma 1, Lemma 3 and Lemma 4, we have the following theorem.

Theorem 1. *Under Assumption 1, the local minima given in Lemma 1 are spurious if* $R(\left(W^l, \mathbf{b}^l\right)_{l=1}^L) > 0$ *for squared or cross-entropy loss.*

3.2 An Auxiliary Lemma

At each location $j \in [M]$, samples are organized into groups according to v_i^j $(i \in [N])$. v_i^js in each group are equal. Let I_q^j be the set of indices of samples in the qth group of the jth location. We have the following lemma to show that $\sum_{i\in I_q^j} u_i$ cannot equal to zero for all groups and all locations.

Lemma 5. *Under Assumption 1, for CNNs with squared or cross-entropy loss, if* $R(\left(W^l, \mathbf{b}^l\right)_{l=1}^L) > 0$*, then there must exist some* $j \in [M]$ *and* $q \in [g_j]$ *such that* $\sum_{i\in I_q^j} u_i \neq 0$*.*

Lemma 5 will be used by Lemma 2 to find data split that satisfies (9).

We will prove Lemma 5 by induction. The number of groups g_j at each location and the value of v_i^j for each group can be arbitrary. After discussing some special cases, we will prove that if Lemma 5 holds for m samples, then for all possible configurations of current groups and all ways of adding a new sample, it also holds for $m+1$ samples. The proof is given in Appendix D.5.

More auxiliary lemmas and their proofs are also given in appendices.

4 An Example of Nontrivial Spurious Local Minima

In Lemma 1, the subnetwork is only required to minimize the loss in (7) and no concrete construction is given. In this section, we will construct an example of nontrivial local minimum in which some

neurons are inactive, which can serve as the subnetwork in Lemma 1. By Theorem 1, the associated local minimum for the CNN that contains this subnetwork is spurious. In comparison, in He et al. (2020); Ding et al. (2019); Goldblum et al. (2020); Liu et al. (2021) for deep ReLU networks, spurious local minima are trivial since all ReLU neurons are active and deep neural networks are reduced to linear predictors.

The idea is to split data samples into two groups using a hyperplane that is perpendicular to an axis, and then fit each group of samples using different predictors. We design a CNN architecture with appropriate parameters to generate these two predictors, and show that perturbating parameters will increase or at least maintain the loss.

Given data samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, we use a hyperplane with normal $\mathbf{p}$ to split the data samples. Choose any index $k \in [d_x]$, we set $\mathbf{p} = (0, 0, \cdots, 1, 0, \cdots, 0)^T$ where only the kth component of $\mathbf{p}$ is 1. Without loss of generality, assume the kth component of input $\mathbf{x}$ is in located in its first channel. Let $I = \left\{i | x_i^k = \max_{j\in[N]} x_j^k\right\}$, i.e., I is the set of indices of samples having the largest kth component. Choose one element i^* in set I, and denote by j^* the index of any sample with the largest x^k among x_n^k $(n \in [N], n \notin I)$, then we set

$$c_p = -\frac{1}{2}\left(x_{i_*}^k + x_{j_*}^k\right). \tag{15}$$

For any point $\mathbf{x}$ on the positive side of hyperplane $\mathbf{p}$, we have

$$\sigma(\mathbf{p}\cdot\mathbf{x} + c_p) = \sigma(x^k - \frac{1}{2}(x_{i^*}^k + x_{j^*}^k)) = x^k - \frac{1}{2}(x_{i^*}^k + x_{j^*}^k) > 0. \tag{16}$$

We fit the two groups of samples in different ways. For samples $\{\mathbf{x}_j | j \in [N], j \notin I\}$, we use a subnetwork with parameter matrix W_2 to fit them, i.e., the output of this subnetwork is

$$\mathbf{o}_j = W_2\hat{\mathbf{x}}_j,\ \forall j \in [N], j \notin I, \tag{17}$$

where $\hat{\mathbf{x}}_j$ is the augmented input.

We use a subnetwork with parameter matrix W_1 to fit data samples $\{\mathbf{x}_j | j \in I\}$, and W_1 is given by

$$W_1(i,\cdot)\hat{\mathbf{x}}_j = W_2(i,\cdot)\hat{\mathbf{x}}_j + \lambda_i(\mathbf{p}^T\mathbf{x_j} + c_p),\ j \in I,\ i \in [d_y]. \tag{18}$$

Then, the output is

$$\mathbf{o}_j = W_1\hat{\mathbf{x}}_j = W_2\hat{\mathbf{x}}_j + (\mathbf{p}\cdot\mathbf{x}_j + c_p)\cdot\left(\lambda_1, \lambda_2, \ldots, \lambda_{d_y}\right)^T,\ j \in I. \tag{19}$$

$\{\lambda_i,\ i \in [d_y]\}$ are constants determined as follows. If there is only one sample $\mathbf{x}_{i^*}$ in set I with target output $\mathbf{y}_{i^*}$, there must exist $\tilde{\mathbf{y}}_{i^*}$ such that $l\left(\tilde{\mathbf{y}}_{i^*}, \mathbf{y}_{i^*}\right) = 0$. Let $\mathbf{o}_{i^*} = \tilde{\mathbf{y}}_{i^*}$, then we have $\tilde{y}_{i^*}(j) = W_2(j,\cdot)\hat{\mathbf{x}}_{i^*} + \lambda_j\left(\mathbf{p}\cdot\mathbf{x}_{i^*} + c_p\right), \forall j \in [d_y]$, which leads to

$$\lambda_j = \frac{\tilde{y}_{i^*}(j) - W_2(j,\cdot)\hat{\mathbf{x}}_{i^*}}{x_{i^*}^k + c_p},\ \forall j \in [d_y]. \tag{20}$$

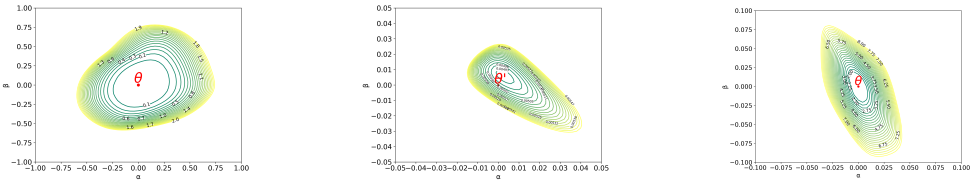
If there are multiple elements in set I, the parameters $\{\lambda_j, j \in [d_y]\}$ are determined by

$$\{\lambda_j,\ j \in [d_y]\} = \text{argmin}_{\{\mu_j,\ j\in[d_y]\}} \sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\mu_1, \mu_2, \cdots, \mu_{d_y}\right)^T, \mathbf{y}_i\right). \tag{21}$$

Concatenating the two subnetworks, we obtain a full CNN. The following theorem gives an informal description of its architecture and parameters. The formal description will be given in Appendix A.

Theorem 2. *(informal). Given a CNN with two subnetworks, its parameters are set as follows. The parameter matrix W_1 of the first subnetwork is set such that the kth component of sample $\mathbf{x}_i$ $(i \in I)$ is forward propagated to the final fully connected layer parameterized by $\{\lambda_j, j \in [d_y]\}$, and that of sample $\mathbf{x}_i$ $(i \notin I)$ is blocked. The parameter matrix W_2 of the second subnetwork is optimized to minimize the empirical loss for samples $\{\mathbf{x}_j | j \in [N], j \notin I\}$. Then, the point $\theta = \left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is a local minimum in parameter space.*

For this local minimum, the ReLU units in the first subnetwork that correspond to the kth component of input are only activated by samples $\mathbf{x}_j$ $(j \in I)$ and are inactive for remaining samples. Thus, the local minimum $\left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is nontrivial for which different activation patterns exist and the resulting predictor is nonlinear.

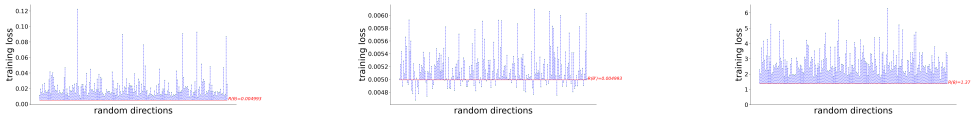


(a) level sets around θ in Lemma 1

(b) level sets around θ' in Lemma 3

(c) level sets around θ in Theorem 2

Figure 1: The visualization of level sets of empirical loss around specific locations in parameter space. (a) and (b) demonstrate that θ is a spurious local minimum since $R(\theta') = R(\theta)$. (c) shows that θ constructed in Theorem 2 is a local minimum.



(a) variation of empirical loss in random directions around θ in Lemma 1

(b) variation of empirical loss in random directions around θ' in Lemma 3

(c) variation of empirical loss in random directions around θ in Theorem 2

Figure 2: The variation of empirical loss in 200 random directions around specific locations in parameter space. In (b), some directions have losses lower than $R(\theta') = R(\theta)$. (a) and (b) together demonstrate that θ is a spurious local minimum. (c) shows that θ constructed in Theorem 2 is a local minimum.

5 Experimental Results

We conduct experiments on CIFAR-10 image set to verify the correctness of Theorem 1 and Theorem 2. We use two approaches to show the existence of spurious local minima. The first one is to visualize the loss landscape around local minima θ using the technique given in Li et al. (2018). Given two random directions δ and η, we compute the empirical losses $R(\theta + \alpha\delta + \beta\eta)$ at grid points in the two-dimensional plane specified by δ and η, where (α, β) are the coodinates of each grid point. Then, the level sets of empirical loss are depicted using empirical losses at these grid points. The second approach is to compute the empirical losses around θ along many random directions and see whether losses lower than $R(\theta)$ exist. Experimental details are given in Appendix C.

Figure 1 shows the results of the level set visualization approach. Figure 2 shows the results of the random direction approach. Fig.1(a) and (b) demonstrate that the local minimum θ constructed in Lemma 1 is spurious since $R(\theta') = R(\theta)$ and local minimum with loss lower than $R(\theta')$ exists. Fig. 1(c) shows that the point θ constructed in Theorem 2 is a local minimum.

6 Conclusion

We have proved that convolutional layers can introduce spurious local minima in the loss landscape of deep convolutional neural networks with squared loss or cross-entropy loss. To show this, we developed new techniques to solve the challenges introduced by convolutional layers. We solved a combinatorial problem to demonstrate that a split of outputs of data samples is always possible somewhere in feature maps. In this combinatorial problem, we overcame the difficulty of arbitrary groupings of outputs of data samples caused by limited receptive fields and arbitrary activation status of hidden neurons. We also solved the tied parameters problem, giving perturbations of filters and biases to decrease the training loss that affect only the output of a single neuron in the feature map.

In future work, we plan to explore the existence of spurious local minima for CNNs with more general loss functions.

REFERENCES

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, 2014.

Tian Ding, Dawei Li, and Ruoyu Sun. Spurious local minima exist for almost all over-parameterized neural networks. *Optimization online*, 2019.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A. Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, 2018.

Simon S. Du and Jason D. Lee. On the power of over-parametrization in neural networks with quadratic activation. In *International Conference on Machine Learning*, 2018.

Simon S. Du, Jason D. Lee, Yuandong Tian, Barnabás Póczos, and Aarti Singh. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. In *International Conference on Machine Learning*, 2018.

C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. In *International Conference on Learning Representations*, 2017.

K. Fukumizu and S. Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13:317–327, 2000.

Kenji Fukumizu, Shoichiro Yamaguchi, Yoh ichi Mototake, and Mirai Tanaka. Semi-flat minima and saddle points by embedding neural networks to overparameterization. In *Advances in Neural Information Processing Systems*, 2019.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018.

R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, 2016.

M. Goldblum, J. Geiping, A. Schwarzschild, M. Moeller, and T. Goldstein. Truth or backpropaganda? an empirical investigation of deep learning theory. In *International Conference on Learning Representations*, 2020.

Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations*, 2015.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. In *International Conference on Learning Representations*, 2017.

Fengxiang He, Bohan Wang, and Dacheng Tao. Piecewise linear activations substantially shape the loss surfaces of neural networks. In *International Conference on Learning Representations*, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.

Gao Huang, Zhuang Liu, and Laurens van der Maaten. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

K. Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, 2016.

Abbas Kazemipour, Brett Larsen, and Shaul Druckmann. No spurious local minima in deep quadratic networks. *arXiv preprint arXiv:2001.00098*, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Thomas Laurent and James H. von Brecht. Deep linear networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning*, 2018.

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.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, 2018.

Qianli Liao and Tomaso Poggio. Theory of deep learning ii: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833*, 2017.

Bo Liu. Spurious local minima are common for deep neural networks with piecewise linear activations. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2022. doi: 10.1109/TNNLS.2022.3204319.

Bo Liu, Zhaoying Liu, Ting Zhang, and Tongtong Yuan. Non-differentiable saddle points and sub-optimal local minima exist for deep relu networks. *Neural Networks*, 144:75–89, 2021.

Haihao Lu and Kenji Kawaguchi. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*, 2017.

Christian H.X. Ali Mehmeti-Gopel, David Hartmann, and Michael Wand. Ringing relus: Harmonic distortion analysis of nonlinear feedforward networks. In *International Conference on Learning Representations*, 2021.

Q. Nguyen and M. Hein. Optimization landscape and expressivity of deep cnns. In *International Conference on Machine Learning*, 2018.

Maher Nouiehed and Meisam Razaviyayn. Learning deep models: Critical points and local openness. *arXiv preprint arXiv:1803.02968*, 2018.

I. Safran and O. Shamir. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, 2018.

Arsalan Sharifnassab, Saber Salehkaleybar, and S. Jamaloddin Golestani. Bounds on over-parameterization for guaranteed existence of descent paths in shallow relu networks. In *International Conference on Learning Representations*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of overparameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2): 742–769, 2019.

Grzegorz Swirszcz, Wojciech Marian Czarnecki, and Razvan Pascanu. Local minima in training of deep networks. *arXiv preprint arXiv:1611.06310*, 2016.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. In *International Conference on Learning Representations*, 2018.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. In *International Conference on Learning Representations*, 2019.

Li Zhang. Depth creates no more spurious local minima. *arXiv preprint arXiv:1901.09827*, 2019.

Yaoyu Zhang, Yuqing Li, Zhongwang Zhang, Tao Luo, and Zhi-Qin John Xu. Embedding principle: A hierarchical structure of loss landscape of deep neural networks. *Journal of Machine Learning*, 1:60–113, 2022.

Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. In *International Conference on Learning Representations*, 2018.

A AN EXAMPLE OF NONTRIVIAL LOCAL MINIMA FOR CNNS

The following theorem gives the architecture and parameters $\left(W^l, \mathbf{b}^l\right)_{l=1}^L$ for a CNN, and shows that $\theta := \left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is a local minimum. For ease of presentation, we ignore pooling layers at first.

Theorem 3. *(formal version of Theorem 2). Given a CNN, assume the number of filters in each convolutional layer satisfies $T_l \geq 2$ $(l \in [L-1])$, and the size of each convolutional layer satisfies $n_l - P_{l-1} \geq d_y$ $(l \in [L-1])$. Its parameters are given as follows. The weight matrix of the first filter in the first convolutional layer is*

$$W^1(1:P_0, 1:P_0) = I_{d_{P_0 \times P_0}}, \tag{22}$$

where $I_{d_{P_0 \times P_0}}$ is the identity matrix of size $P_0 \times P_0$, P_0 is the number of patches in the input used for convolution. The corresponding bias is

$$b_1^1 = c_p. \tag{23}$$

The filters and biases of higher convolution layers are set as

$$W^l(1:P_{l-1}, 1:P_{l-1}) = I_{d_{P_{l-1} \times P_{l-1}}}, \;\; b_1^l = 0, \;\; l \in \{2, 3, \cdots, L-1\} \tag{24}$$

to forward propagate the first feature map of $\boldsymbol{o}^1(\boldsymbol{x}_i)$ unchanged.

The subnetwork with parameter matrix W_2 is formed by the part of CNN that removes the first filter and all activation units in each layer, and its output is given as follows. For all $j \in [N]$ and $j \notin I$,

$$\begin{aligned}\mathbf{o}_j = W_2 \hat{\mathbf{x}}_j :=& W^L(\cdot, P_{L-2}+1:n_{L-1})[W^{L-1}(P_{L-2}+1:n_{L-1}, P_{L-3}+1:n_{L-2}) \cdots \\ & (W^1(P_0+1:n_1, \cdot)\mathbf{x}_j + b^1(P_0+1:n_1)) + \cdots + b^{L-1}(P_{L-2}+1:n_{L-1})] + \mathbf{b}^L.\end{aligned} \tag{25}$$

W_2 is optimized to minimize $\sum_{\substack{i \in [N], \\ i \notin I}} l\left(W_2 \hat{\mathbf{x}}_i, \boldsymbol{y}_i\right)$.

The final fully connected layer is set as

$$W^L\left(i, 1:P_{L-2}\right) = \left(0, \cdots, \lambda_i, \cdots, 0\right), \; i \in [d_y], \tag{26}$$

where only the kth component connecting to the first subnetwork is nonzero and equals to λ_i.

Sufficiently large positive constants c_l $(l \in [L-1])$ are added to the biases of each convolutional layer in the second subnetwork such that the involved ReLU units are all activated. The biases of top fully connected layer b_i^L $(i \in [d_y])$ are then substituted by $b_i^L - \Delta o^L(i)$, where $\Delta o^L(i)$ is recursively computed as

$$\begin{aligned}&\Delta o^1(i) = c_1, \;\; i \in [P_0+1:n_1], \\ &\Delta o^l(i) = \sum_{j \in [P_{l-2}+1:n_{l-1}]} W^l(i,j)\Delta o^{l-1}(j) + c_l, \; i \in [P_{l-1}+1:n_l], \; l \in \{2, 3, \cdots, L-1\}, \\ &\Delta o^L(i) = \sum_{j \in [P_{L-2}+1:n_{L-1}]} W^L(i,j)\Delta o^{L-1}(j), \; i \in [d_y].\end{aligned} \tag{27}$$

All remaining parameters are set to zeros. Then, the point $\theta = \left(W^l, \mathbf{b}^l\right)_{l=1}^L$ specified above is local minimum in parameter space.

If pooling layers are used in CNNs, we can adjust the parameters in Theorem 3 to reflect the effects of parameter matrices of pooing layers. However, the adjusted $\left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is still a local minimum.

The condition for the number of filters is used to contain two subnetworks. The condition for the size of each convolutional layers is used to make sure θ presented in Theorem 3 meets the full rank requirement. Because the size of each convolutional layer satisfies $n_l - P_{l-1} \geq d_y$ $(l \in [L-1])$, the matrix $W_2 \in \mathbb{R}^{d_y \times d_x}$ for the second predictor is full rank around θ (see the proof of Theorem 3 in Appendix D.9). These conditions are easily satisfied by most CNNs used in practice.

B MORE AUXILIARY LEMMAS

Denote $u_i := \frac{\partial l(\mathbf{o}(\mathbf{x}_i),\mathbf{y}_i)}{\partial o(1)}$ and $o_i := o^{L,i}(1)$, we have the following lemma for squared and cross-entropy loss.

Lemma 6. *For convolutional neural networks with squared loss or cross-entropy loss, the followings can be achieved through perturbating $W^L(1,\cdot)$ under Assumption 1, where o_i' and u_i' denote, respectively, o_i and u_i after perturbation.*

1. $o_i' \neq o_j'$ if $o_i = o_j$ $(i,j \in [N],\ i \neq j)$.

2. $u_i' \neq 0$ if $u_i = 0$ $(i \in [N])$.

3. $u_i' \neq u_j'$ or $u_i' + u_j' \neq 0$ if $u_i = u_j$ or $u_i + u_j = 0$, respectively, $(i,j \in [N],\ i \neq j)$.

4. $\boldsymbol{u} = (u_1, u_2, \cdots, u_N)^T \neq \boldsymbol{0}$ if $R = \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{o}_i, \mathbf{y}_i) > 0$.

5. Nonzero gaps between o_i's or u_i's can still exist after subsequent perturbations.

The maintenance of nonzero gaps between o_i's or u_i's guarantees that $o_i' \neq o_j'$ or $u_i' \neq u_j'$ etc. still hold after subsequent perturbations.

In the following, we absorb biases into weights and the CNN output is written as $\mathbf{o}_i = W^L I^{L-1,i} W^{L-1} \cdots I^{1,i} W^1 \mathbf{x}_i$. Here, we omit the hat on augmented variables for notational simplicity, and the exact meaning of involved symbols is clear from context. The following two lemmas will be used to prove $u_i' \neq u_j'$ or $u_i' + u_j' \neq 0$ $(i,j \in [N],\ i \neq j)$ in Lemma 6 for cross-entropy loss.

Lemma 7. *For cross-entropy loss, if $u_i = u_j$ for two samples $\mathbf{x}_i, \mathbf{x}_j$ $(i \neq j)$ and $I^{L-1,i} W^{L-1} \cdots W^1 \mathbf{x}_i = \alpha I^{L-1,j} W^{L-1} \cdots I^{1,j} W^1 \mathbf{x}_j$ $(\alpha \neq 0, \alpha \neq 1)$, then under Assumption 1, $u_i' \neq u_j'$ can be achieved through perturbating $W^L(1,\cdot)$.*

Lemma 8. *For cross-entropy loss, if $u_i + u_j = 0$ for two samples $\mathbf{x}_i, \mathbf{x}_j$ $(i \neq j)$ and $I^{L-1,i} W^{L-1} \cdots W^1 \mathbf{x}_i = \alpha I^{L-1,j} W^{L-1} \cdots I^{1,j} W^1 \mathbf{x}_j$ $(\alpha \neq 0, \alpha \neq 1)$, then under Assumption 1, $u_i' + u_j' \neq 0$ can be achieved through perturbating $W^L(1,\cdot)$.*

C EXPERIMENTAL DETAILS

We use CIFAR-10 image set to train CNNs, which consists of 10 classes and 50000 training images of size 32×32. The CNN used in our experiments has 7 convolutional layers, with the number of channels being $64, 64, 128, 128, 256, 256, 256$, respectively, and no pooling layers are used. The convolution filters are all 3×3. Each convolutional layer is followed by a ReLU layer. The subnetwork in Lemma 1 is trained by Adam optimizer with 150 epochs, with a learning rate of 0.001 and weight decay of 0.0005. The subnetwork W_2 in Theorem 3 is trained by Adam optimizer with 500 epochs, with a learning rate of 0.0001 and weight decay of 0.0005. Both subnetworks use the Kaiming initialization (He et al. (2015)). The biggest threshold η in Lemma 2 is found to be 6.8915.

D PROOFS

For the proofs in the following sections, we assume that the perturbation of network parameters for a differentiable local minimum $\theta = \left(W^l, \mathbf{b}^l\right)_{l=1}^{L}$ is very small such that the activation patterns $I^{l,i}$ $(l \in [L-1], i \in [N])$ will keep constant.

D.1 PROOF OF LEMMA 1

Proof. When parameters are perturbated as $\left(W^l, \mathbf{b}^l\right)_{l=1}^{L} \longrightarrow \left(W^l + \delta W^l, \mathbf{b}^l + \delta \mathbf{b}^l\right)_{l=1}^{L}$, the output after perturbation is as follows,

$$\begin{aligned} \mathbf{o}'(\mathbf{x}_i) &= \left(\hat{W}^L + \delta \hat{W}^L\right) \hat{I}^{L-1,i} \left(\hat{W}^{L-1} + \delta \hat{W}^{L-1}\right) \hat{I}^{L-2,i} \cdots \left(\hat{W}^1 + \delta \hat{W}^1\right) \hat{\mathbf{x}}_i \\ &= (\hat{W}^L \hat{I}^{L-1,i} \hat{W}^{L-1} \hat{I}^{L-2,i} \cdots \hat{W}^1 + \delta F^i) \hat{\mathbf{x}}_i, \quad \forall i \in [N], \end{aligned} \tag{28}$$

where $\delta F^i = \delta\hat{W}^L\hat{I}^{L-1,i}\hat{W}^{L-1}\hat{I}^{L-2,i}\cdots\hat{W}^1 + \hat{W}^L\hat{I}^{L-1,i}\delta\hat{W}^{L-1}\hat{I}^{L-2,i}\cdots\hat{W}^1 + \cdots + \delta\hat{W}^L\hat{I}^{L-1,i}\delta\hat{W}^{L-1}\hat{I}^{L-2,i}\cdots\delta\hat{W}^1$.

The training loss after perturbation is

$$\begin{aligned} R((W^l+\delta W^l, \mathbf{b}^l+\delta\mathbf{b}^l)_{l=1}^L) &= \frac{1}{N}\sum_{i=1}^N l\left(\mathbf{o}'\left(\mathbf{x}_i\right), \mathbf{y}_i\right) \\ &= \frac{1}{N}\sum_{i=1}^N l\left(\hat{W}^L\hat{I}^{L-1,i}\hat{W}^{L-1}\hat{I}^{L-2,i}\cdots\hat{W}^1\hat{\mathbf{x}}_i + \delta F^i\hat{\mathbf{x}}_i, \mathbf{y}_i\right). \end{aligned} \tag{29}$$

Note that the matrix $\hat{W}^L\hat{I}^{L-1,i}\hat{W}^{L-1}\cdots\hat{W}^1$ is assumed to be full rank for all $i \in [N]$ in a small neighborhood around $\left(\left(W^l, \mathbf{b}^l\right)_{l=1}^{L-3}, W^L(\cdot, 2P_{L-2}+1 : end), \mathbf{b}^L\right)$, then when minimizing $\tilde{R}(\left(W^l, \mathbf{b}^l\right)_{l=1}^{L-3}, W^L(\cdot, 2P_{L-2}+1 : end), \mathbf{b}^L) := \frac{1}{N}\sum_{i=1}^N l\left(\hat{W}^L\hat{I}^{L-1,i}\hat{W}^{L-1}\hat{I}^{L-2,i}\cdots\hat{W}^1\hat{\mathbf{x}}_i, \mathbf{y}_i\right)$ in (7), the space of $\mathbf{o}\left(\mathbf{x}_i\right)$ has been fully explored by $\hat{W}^L\hat{I}^{L-1,i}\hat{W}^{L-1}\cdots\hat{W}^1$ in the neighborhood during minimization. Therefore, $R((W^l+\delta W^l, \mathbf{b}^l+\delta\mathbf{b}^l)_{l=1}^L)$ cannot be lower than $R((W^l, \mathbf{b}^l)_{l=1}^L)$ despite the perturbation in δF^i, and $\left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is thus a local minimum in parameter space.

□

D.2 Proof of Lemma 2

Proof. At a location $j \in [M]$, if there exist some groups in the ordered list of v_i^j $(i \in [N])$ such that $\sum_{i\in I_q^j} u_i \neq 0$, and suppose $G_1 = \left\{v_{m_q+1}^j, v_{m_q+2}^j, \cdots, v_{m_{q+1}}^j\right\}$ is the first such group in the ordered list, we can set $\eta = \frac{1}{2}\left(v_{m_{q+1}+1}^j + v_{m_{q+1}}^j\right)$, which is the midpoint of the gap between group G_1 and the next group, and set $n = m_{q+1}$. We then have $\sum_{i=1}^n u_i = \sum_{i=1}^{m_q} u_i + \sum_{i\in G_1} u_i = \sum_{i\in G_1} u_i \neq 0$. The requirements in (9) are satisfied and we have found a split for data samples.

If for every group $q \in [g_j]$ at location j, there is $\sum_{i\in I_q^j} u_i = 0$, we have to explore other locations in feature maps to see whether there exist such splits. According to Lemma 5, $\sum_{i\in I_q^j} u_i = 0$ cannot happen for every group q and every location j, and there must exist some locations where $\sum_{i\in I_q^j} u_i \neq 0$ for some groups. As a result, we can split data samples as before at these locations.

There may be more than one location where such data split exists, and we choose the one with the biggest threshold η and use it in the perturbation of training loss. Let h be the location having the biggest split threshold, we then have

$$\forall j \in [M], j \neq h, \quad \sum_{i\in\left\{1,2,\cdots,N \mid v_i^j > \eta\right\}} u_i = 0. \tag{30}$$

□

D.3 Proof of Lemma 3

Proof. There are two possible cases for $\eta : \eta \geq -b_1^{L-3}$ or $\eta < -b_1^{L-3}$, which will be treated differently in the following.

In order to compare the training losses at θ and θ', we need to compute the output $\mathbf{o}'\left(\mathbf{x}_i\right)$ obtained by θ'. We will give the computation of its first component $o_1'\left(\mathbf{x}_i\right)$ in detail. We only need to compute those terms in $o_1'\left(\mathbf{x}_i\right)$ that are affected by the parameter settings in (11) and (12), and denote by $\tilde{o}'\left(\mathbf{x}_i\right)$ the sum of such terms.

If $\eta \geq -b_1^{L-3}$, only the first and the second feature maps in layers $L-1$ and $L-2$ are involved. Let $\mathbf{o}'^{L-3,i} := \sigma(W^{L-3}\left(1 : P_{L-4}, \cdot\right)\mathbf{o}^{L-4,i} + b_1'^{L-3}\mathbf{1}_{P_{L-4}})$, where $\mathbf{1}_{P_{L-4}}$ is the vector of size P_{L-4} whose components are all 1s. $\mathbf{o}'^{L-3,i}$ is the output of the first feature map at layer $L-3$ using

the new bias $b_1'^{L-3}$. Since $W^{L-3}(j,\cdot)\,\mathbf{o}^{L-4,i}+b_1'^{L-3}\geq 0$ for all $i\in[N],\ j\in[P_{L-4}]$, there is $\mathbf{o}'^{L-3,i}=W^{L-3}(1:P_{L-4},\cdot)\,\mathbf{o}^{L-4,i}+b_1'^{L-3}\mathbf{1}_{P_{L-4}}$. Then, the output $\tilde{o}'(\mathbf{x}_i)$ is

$$\begin{aligned}\tilde{o}'(\mathbf{x}_i)=&W'^{L}(1,1:P_{L-2})\\&\sigma\big(W'^{L-1}(1:P_{L-2},1:P_{L-3})\\&\sigma\big(W'^{L-2}(1:P_{L-3},1:P_{L-4})\,\mathbf{o}'^{L-3,i}+b_1'^{L-2}\mathbf{1}_{P_{L-3}}\big)+b_1'^{L-1}\mathbf{1}_{P_{L-2}}\big)+\\&W'^{L}(1,P_{L-2}+1:2P_{L-2})\\&\sigma\big(W'^{L-1}(P_{L-2}+1:2P_{L-2},P_{L-3}+1:2P_{L-3})\\&\sigma\big(W'^{L-2}(P_{L-3}+1:2P_{L-3},1:P_{L-4})\,\mathbf{o}'^{L-3,i}+b_2'^{L-2}\mathbf{1}_{P_{L-3}}\big)+b_2'^{L-1}\mathbf{1}_{P_{L-2}}\big)+b_1'^{L}.\end{aligned}\tag{31}$$

For notational simplicity, denote $W'^{L}(1,i)\,(i\in[P_{L-2}])$ as w^i, and note that $v_i^j:=W^{L-3}(j,\cdot)\,\mathbf{o}^{L-4,i},\ (j\in[P_{L-4}])$. Using (11), we have

$$\begin{aligned}\tilde{o}'(\mathbf{x}_i)=&\sum_{j=1}^{P_{L-2}}[w^j\sigma\Big(-\sigma\Big((-v_i^j-b_1'^{L-3})+\eta+b_1'^{L-3}\Big)+\eta+b_1^{L-3}\Big)+\\&\qquad w^j\sigma\Big(\sigma\Big((v_i^j+b_1'^{L-3})-\eta-b_1'^{L-3}\Big)+\eta+b_1^{L-3}\Big)-w_j\big(\eta+b_1^{L-3}\big)]+b_1^L\\=&\sum_{j=1}^{P_{L-2}}w^j\cdot z_j'+b_1^L,\end{aligned}\tag{32}$$

where

$$\begin{aligned}z_j':=&\sigma\Big(-\sigma\Big((-v_i^j-b_1'^{L-3})+\eta+b_1'^{L-3}\Big)+\eta+b_1^{L-3}\Big)+\\&\sigma\Big(\sigma\Big((v_i^j+b_1'^{L-3})-\eta-b_1'^{L-3}\Big)+\eta+b_1^{L-3}\Big)-\big(\eta+b_1^{L-3}\big).\end{aligned}\tag{33}$$

We also denote

$$z_j:=\sigma(v_i^j+b_1^{L-3}),\tag{34}$$

and thus the partial output obtained from θ is

$$\tilde{o}(\mathbf{x}_i)=\sum_{j=1}^{P_{L-2}}w^j\cdot z_j+b_1^L.\tag{35}$$

There are three possible cases for the value of v_i^j. Note that we already have $\eta+b_1^{L-3}\geq 0$ if $\eta\geq -b_1^{L-3}$.

1) $v_i^j\geq\eta$.
In this case, we have $v_i^j-\eta\geq 0$ and $v_i^j+b_1^{L-3}\geq v_i^j-\eta\geq 0$, then

$$\begin{aligned}z_j'&=\sigma(\eta+b_1^{L-3})+\sigma(v_i^j-\eta+\eta+b_1^{L-3})-\big(\eta+b_1^{L-3}\big)\\&=v_i^j+b_3^{L-3}=z_j>0,\ \ \forall j\in[P_{L-2}].\end{aligned}\tag{36}$$

2) $-b_1^{L-3}\leq v_i^j<\eta$.
We have $-v_i^j+\eta>0$. By (33) and $v_i^j+b_1^{L-3}\geq 0$, we then have

$$\begin{aligned}z_j'&=\sigma\Big(-\Big(-v_i^j+\eta\Big)+\eta+b_1^{L-3}\Big)+\sigma\big(\eta+b_1^{L-3}\big)-\big(\eta+b_1^{L-3}\big)\\&=\sigma(v_i^j+b_1^{L-3})=z_j\geq 0,\ \ \forall j\in[P_{L-2}].\end{aligned}\tag{37}$$

3) $v_i^j<-b_1^{L-3}$.
In this case, we have $v_i^j<\eta$, $v_i^j+b_1^{L-3}<0$, then

$$\begin{aligned}z_j'&=\sigma\Big(v_i^j-\eta+\eta+b_1^{L-3}\Big)+\sigma\big(\eta+b_1^{L-3}\big)-\big(\eta+b_1^{L-3}\big)\\&=\sigma\Big(v_i^j+b_1^{L-3}\Big)=0=z_j,\ \ \forall j\in[P_{L-2}].\end{aligned}\tag{38}$$

In all three possible cases, we have obtained $z'_j = z_j, \forall j \in [P_{L-2}]$. Therefore, $\tilde{o}'(\mathbf{x}_i) = \sum_{j=1}^{P_{L-2}} w^j \cdot z_j + b_1^L = \tilde{o}(\mathbf{x}_i), \ \forall i \in [N]$. We can obtain $o_k(\mathbf{x}_i) = o'_k(\mathbf{x}_i)$ $(\forall i \in [N], \forall k \in [d_y], k \neq 1)$ for remaining output components since $W'^L(2 : d_y, 1 : 2P_{L-2}) = 0$. The outputs are thus not changed, so does the training loss.

We now discuss the case of $\eta < -b_1^{L-3}$. Denote $w^i := W^L(1, 2P_{L-2} + i), \forall i \in [P_{L-2}]$. The output $\tilde{o}'(\mathbf{x}_i)$ that is affected by the parameter setting in (12) is as follows,

$$
\begin{aligned}
\tilde{o}'(\mathbf{x}_i) =& W'^L(1, 1 : P_{L-2}) \\
& \sigma\big(W'^{L-1}(1 : P_{L-2}, 1 : P_{L-3}) \\
& \sigma\big(W'^{L-2}(1 : P_{L-3}, 1 : P_{L-4})\, \mathbf{o}'^{L-3,i} + b_1'^{L-2}\mathbf{1}_{P_{L-3}}\big) + b_1'^{L-1}\mathbf{1}_{P_{L-2}}\big) + \\
& W'^L(1, P_{L-2} + 1 : 2P_{L-2}) \\
& \sigma\big(W'^{L-1}(P_{L-2} + 1 : 2P_{L-2}, P_{L-3} + 1 : 2P_{L-3}) \\
& \sigma\big(W'^{L-2}(P_{L-3} + 1 : 2P_{L-3}, 1 : P_{L-4})\, \mathbf{o}'^{L-3,i} + b_2'^{L-2}\mathbf{1}_{P_{L-3}}\big) + b_2'^{L-1}\mathbf{1}_{P_{L-2}}\big) + \\
& W'^L(1, 2P_{L-2} + 1 : 3P_{L-2}) \\
& \sigma\big(W'^{L-1}(2P_{L-2} + 1 : 3P_{L-2}, 2P_{L-3} + 1 : 3P_{L-3}) \\
& \sigma\big(W'^{L-2}(2P_{L-3} + 1 : 3P_{L-3}, 1 : P_{L-4})\, \mathbf{o}'^{L-3,i} + b_3'^{L-2}\mathbf{1}_{P_{L-3}}\big) + b_3'^{L-1}\mathbf{1}_{P_{L-2}}\big) + b_1'^L.
\end{aligned} \tag{39}
$$

Using (12), we have

$$
\begin{aligned}
\tilde{o}'(\mathbf{x}_i) = \sum_{j=1}^{p_{L-2}} w^j [& \sigma\left(\sigma\left(v_i^j + b_1'^{L-3} - \eta - b_1'^{L-3}\right)\right) \\
& -\sigma\left(\sigma\left(v_i^j + b_1'^{L-3} - \eta - b_1'^{L-3}\right)\right) \\
& +\sigma\left(\sigma\left(v_i^j + b_1'^{L-3} - \eta - b_1'^{L-3}\right) + \eta + b_1^{L-3}\right)] + b_1'^L. \\
= & \sum_{j=1}^{p_{L-2}} w^j z'_j + b_1^L,
\end{aligned} \tag{40}
$$

where

$$
\begin{aligned}
z'_j :=& \sigma\left(\sigma\left(v_i^j + b_1'^{L-3} - \eta - b_1'^{L-3}\right)\right) \\
& -\sigma\left(\sigma\left(v_i^j + b_1'^{L-3} - \eta - b_1'^{L-3}\right)\right) \\
& +\sigma\left(\sigma\left(v_i^j + b_1'^{L-3} - \eta - b_1'^{L-3}\right) + \eta + b_1^{L-3}\right).
\end{aligned} \tag{41}
$$

There are also three possible cases for the value of v_i^j. Note that we already have $\eta + b_1^{L-3} < 0$ when $\eta < -b_1^{L-3}$.
1) $v_i^j \geq -b_1^{L-3}$.
In this case, using $v_i^j - \eta > 0, \ v_i^j + b_1^{L-3} \geq 0$ and (41), we have

$$
\begin{aligned}
z'_j &= \left(v_i^j - \eta\right) - \left(v_i^j - \eta\right) + \sigma\left(v_i^j + b_1^{L-3}\right) \\
&= v_i^j + b_1^{L-3} = z_j \geq 0, \ \ \forall j \in [p_{L-2}].
\end{aligned} \tag{42}
$$

2) $\eta \leq v_i^j < -b_1^{L-3}$.
Using $v_i^j - \eta \geq 0, \ v_i^j + b_1^{L-3} < 0$, we have

$$
\begin{aligned}
z'_j &= \left(v_i^j - \eta\right) - \left(v_i^j - \eta\right) + \sigma\left(v_i^j + b_1^{L-3}\right) \\
&= 0 = z_j, \ \ \forall j \in [p_{L-2}].
\end{aligned} \tag{43}
$$

3) $v_i^j < \eta$.
Using $v_i^j - \eta < 0,\ v_i^j + b_1^{L-3} < 0$, we have

$$z_j' = \sigma\left(\eta + b_1^{L-3}\right) = 0 = z_j = \sigma\left(v_i^j + b_1^{L-3}\right),\ \ \forall j \in [p_{L-2}]\,. \tag{44}$$

Therefore, in all possible cases of η and v_i^j, the output $o_1'(\mathbf{x}_i) = o_1(\mathbf{x}_i), \forall i \in [N]$. We have $o_k(\mathbf{x}_i) = o_k'(\mathbf{x}_i)$ $(\forall i \in [N], \forall k \in [d_y], k \neq 1)$ for remaining output components. As a result, the training loss satisfies $R(\theta) = R(\theta')$.

□

D.4 Proof of Lemma 4

Proof. We will show that by perturbating θ', the training loss can be decreased. We use different perturbation schemes to show this for different cases of η.

If $\eta \geq -b_1^{L-3}$, we only perturbate $b_2'^{L-2} = -\eta - b_1'^{L-3}$ as

$$b_2'^{L-2} \to b_2'^{L-2} + \delta b = -\eta - b_1'^{L-3} + \delta b. \tag{45}$$

Under this perturbation, the output of each sample is perturbated as follows according to different cases of v_i^j.
1) $v_i^j \geq \eta$.
In this case, after perturbation, by modifying (33) we have

$$z_j' = \sigma\left(\sigma(v_i^j - \eta + \delta b) + \eta + b_1^{L-3}\right).$$

Since η is at the midpoint of a gap between adjacent v_i^js (as indicated in the proof of Lemma 2), there is $v_i^j > \eta$, hence for sufficiently small perturbation $\delta b,\ v_i^j - \eta + \delta b > 0$ and $v_i^j + b_3^{L-3} + \delta b > 0$ hold. Therefore,

$$z_j' = v_i^j + b_1^{L-3} + \delta b = z_j + \delta b,\ \ \forall j \in [P_{L-2}]\,. \tag{46}$$

2) $-b_1^{L-3} \leq v_i^j < \eta$.
Since $v_i^j - \eta + \delta b < 0$, the perturbation δb does not pass through the ReLU activation,

$$z_j' = v_i^j + b_1^{L-3} = z_j,\ \ \forall j \in [P_{L-2}]\,. \tag{47}$$

3) $v_i^j < -b_1^{L-3}$.
In this case, $v_i^j - \eta + \delta b < 0$, then

$$z_j' = 0 = z_j,\ \ \forall j \in [P_{L-2}]\,. \tag{48}$$

Combining the above three cases, it is found that the output $o'(\mathbf{x}_i)$ is perturbated only if $v_i^j > \eta$. Correspondingly, the perturbation of output is

$$\delta o'(\mathbf{x}_i) = \delta\tilde{o}'(\mathbf{x}_i) = \sum_{j=1}^{P_{L-2}} w^j \cdot \delta b \cdot I\left(v_i^j > \eta\right), \tag{49}$$

where $I()$ is the indicator function. Other components of output do not change under this perturbation since $W'^L(2:d_y, 1:2P_{L-2}) = 0$. The training loss is perturbated as

$$\delta R' := R(\theta' + \delta\theta') - R(\theta') = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial l\left(\mathbf{o}'(\mathbf{x}_i), \mathbf{y}_i\right)}{\partial o_1} \cdot \delta o'(\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial l\left(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i\right)}{\partial o_1} \cdot \delta o'(\mathbf{x}_i),$$

here $\mathbf{o}(\mathbf{x}_i) = \mathbf{o}'(\mathbf{x}_i)$ is used to get $\frac{\partial l\left(\mathbf{o}'(\mathbf{x}_i), \mathbf{y}_i\right)}{\partial o_1} = \frac{\partial l(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i)}{\partial o_1}$. Therefore, we have

$$\delta R' = \frac{1}{N}\sum_{i=1}^{N} u_i \cdot \sum_{j=1}^{P_{L-2}} w^j \cdot \delta b \cdot I\left(v_i^j > \eta\right). \tag{50}$$

For all samples $\mathbf{x}_i$ with $v_i^j > \eta$ ($\forall j \in [P_{L-2}]$), by Lemma 2, only at the hth neuron in the first feature map of layer $L-3$, $\sum_{i=1}^n u_i \neq 0$ and $v_i^j > \eta$ ($\forall i \in [n]$) both hold. At other locations in this feature map, because η is the biggest split threshold among all locations, even if $v_i^j > \eta$ holds, there is $\sum_{i \in \left\{1,2,\cdots,N | v_i^j > \eta\right\}} u_i = 0$ (see (10)). We then have

$$\delta R' = \frac{1}{N}\delta b \cdot \sum_{j=1}^{P_{L-2}} w^j \cdot \sum_{i \in \left\{1,2,\cdots,N | v_i^j > \eta\right\}} u_i = \frac{1}{N}\delta b \cdot w^h \cdot \sum_{i=1}^n u_i \tag{51}$$

Setting $\text{Sgn}\left(\delta b\right) = -\text{Sgn}\left(w^h \cdot \sum_{i=1}^n u_i\right)$ if $w^h \neq 0$, where Sgn() is the sign function, we obtain

$$\delta R' < 0. \tag{52}$$

Therefore, the training loss is decreased by the perturbation, resulting in

$$R\left(\theta\right) > R\left(\theta' + \delta\theta'\right). \tag{53}$$

Thus, θ is a spurious local minimum.

If $w^h = 0$, we perturbate it as well by $w^h \to \delta w$, then

$$\delta R' = \frac{1}{N}\delta b \cdot \delta w \cdot \sum_{i=1}^n u_i \tag{54}$$

Setting $\text{Sgn}(\delta b) = -\text{Sgn}\left(\delta w \sum_{i=1}^n u_i\right)$ can still decrease the training loss, and thus the conclusion that θ is a spurious local minimum still holds.

If $\eta < -b_1^{L-3}$, we only perturbate $b_1'^{L-2} = -\eta - b_1'^{L-3}$ as

$$b_1'^{L-2} \longrightarrow b_1'^{L-2} + \delta b = -\eta - b_1'^{L-3} + \delta b. \tag{55}$$

With this perturbation, the output of each sample is perturbated in three different cases according to the value of v_i^j.
1) $v_i^j \geq -b_1^{L-3}$.
In this case, by modifying (41) we have

$$\begin{aligned} z_j' &= \left(v_i^j - \eta + \delta b\right) - \left(v_i^j - \eta\right) + \sigma\left(v_i^j + b_1^{L-3}\right) \\ &= z_j + \delta b, \ \ \forall j \in \left[P_{L-2}\right]. \end{aligned} \tag{56}$$

2) $\eta \leq v_i^j < -b_1^{L-3}$.
In this case, since a gap exists between η and v_i^j when splitting samples with η in Lemma 2, we have $\eta < v_i^j$ and consequently $v_i^j - \eta + \delta b > 0$ for sufficiently small δb. Therefore,

$$z_j' = \left(v_i^j - \eta + \delta b\right) - \left(v_i^j - \eta\right) = \delta b = z_j + \delta b, \ \ \forall j \in \left[P_{L-2}\right]. \tag{57}$$

3) $v_i^j < \eta$.
Using $v_i^j - \eta + \delta b < 0$, we have

$$z_j' = \sigma\left(\eta + b_1^{L-3}\right) = 0 = z_j, \ \ \forall j \in \left[P_{L-2}\right]. \tag{58}$$

The perturbation δb has been blocked by ReLU units.

Only in the first and second cases, the output $o'\left(\mathbf{x}_i\right)$ is affected by the perturbation δb. Thus, we have

$$z_j' = z_j + \delta b, \ \text{ if } v_i^j > \eta, \ \ \forall i \in [N], j \in [P_{L-2}]. \tag{59}$$

The output is perturbated as

$$\delta o'\left(\mathbf{x}_i\right) = \sum_{j=1}^{P_{L-2}} w^j \delta z_j' = \sum_{j=1}^{P_{L-2}} w^j \cdot \delta b \cdot I\left(v_i^j > \eta\right). \tag{60}$$

Remaining output components keep unchanged. The training loss is perturbated as follows,

$$\delta R' = \frac{1}{N}\sum_{i=1}^{N} u_i \delta o'(x_i) = \frac{1}{N}\delta b \cdot \sum_{j=1}^{P_{L-2}} w^j \sum_{i\in\left\{1,2,\cdots,N|v_i^j>\eta\right\}} u_i. \tag{61}$$

Because η is the biggest split threshold among all locations, only for the hth location, there is $\sum_{i\in\{1,2,\cdots,N|v_i^j>\eta\}} u_i \neq 0$. We then have

$$\delta R' = \frac{1}{N}\delta b \cdot w^h \cdot \sum_{i=1}^{n} u_i, \tag{62}$$

which is the same as (51). By setting δb with appropriate sign, we can obtain $\delta R' < 0$ and conclude that θ is a spurious local minimum.

If a perturbation of $W^L(1,\cdot)$ is required to split data samples as shown in Lemma 6, the training loss at θ is perturbated as

$$R(\theta+\delta\theta) = R(\theta) + \frac{\partial R(\theta)}{\partial W^L(1,\cdot)}\delta W^L(1,\cdot) = R(\theta) \tag{63}$$

due to the optimality condition $\frac{\partial R(\theta)}{\partial W^L(1,\cdot)} = \mathbf{0}^T$ since θ is a local minimum. The perturbation $\delta\theta$ only modifies $W^L(1,\cdot)$ and does not change other parameters. Therefore, by starting from $\theta+\delta\theta$ and setting the parameters of the last three layers as in Lemma 3, we can obtain a point θ'' with $R(\theta+\delta\theta) = R(\theta'')$. Thus we have $R(\theta) = R(\theta'')$. We then perturbate the biases of θ'' as in Lemma 4 to obtain a loss lower than $R(\theta)$, which shows that θ is a spurious local minimum.

□

D.5 Proof of Lemma 5

Proof. We prove by induction. We will often perturbate the network parameters to tune o_i or u_i ($i \in [N]$), which does not change empirical loss, as shown in (63).

Let us first consider some special cases.

D.5.1 Special Cases: N = 1 or M = 1

If there is only one data sample, i.e., $N = 1$, there will be only one group at each location. Then, by Lemma 6, we have $\mathbf{u} = (u_1) \neq 0$, thus Lemma 5 holds.

We then consider the case of only one location: $M = 1$. If there are some groups with $\sum_{i\in I_q^1} u_i \neq 0$, Lemma 5 already holds. If there exists a group consisting of a single sample with $u_i = 0$, according to Lemma 6, we can perturbate network parameters to get $u_i' \neq 0$ and thus Lemma 5 holds. Otherwise, all groups have multiple samples and there are $\sum_{i\in I_q^1} u_i = 0$ for all groups, and by $R(\theta) > 0$ and consequently $\mathbf{u} \neq 0$, there must exist some group k such that there exists $i \in I_k^1$ and $u_i \neq 0$. All samples in group k have identical outputs, and by Lemma 6, we can separate this sample with $u_i \neq 0$ from other samples in this group by perturbation, i.e., $o_i' \neq o_j'$ ($\forall j \in I_k^1, j \neq i$). By doing so, we can obtain a isolated sample with $u_i' \neq 0$, and thus Lemma 5 is satisfied.

D.5.2 Induction Hypothesis

Now we discuss the cases for $N \geq 2$ and $M > 1$, and use induction to prove.

In order to give our induction hypothesis, we first give some definitions. For multiple locations ($M > 1$), there are usually a number of groups at each location. Recall that I_q^j denotes the set of indices of samples in the qth group of the jth location. We can select one group with $\sum_{i\in I_{q_j}^j} u_i$ (the sum of u_is in this group) from each location j and form a sequence $\sum_{i\in I_{q_1}^1} u_i,\ \sum_{i\in I_{q_2}^2} u_i, \cdots,\ \sum_{i\in I_{q_M}^M} u_i$.

If all elements in this sequence are equal and nonzero, i.e., $\sum_{i\in I^1_{q_1}} u_i = \sum_{i\in I^2_{q_2}} u_i = \cdots = \sum_{i\in I^M_{q_M}} u_i \neq 0$, we call it a complete and identical sequence of groups (**CI sequence**). If all elements in a sequence are nonzero but not equal, or partial elements in this sequence equal to zero, we call it a **non-CI sequence**.

Given a CI sequence, after adding a sample u_k into the group q_j at each location j, if $\sum_{i\in I^1_{q_1}} u_i = \sum_{i\in I^2_{q_2}} u_i = \cdots = \sum_{i\in I^M_{q_M}} u_i = -u_k$, the sequence becomes $\sum_{i\in I^1_{q_1}} u_i = 0$, $\sum_{i\in I^2_{q_2}} u_i = 0, \cdots$, $\sum_{i\in I^M_{q_M}} u_i = 0$ (now the set $I^j_{q_j}$ $(j \in [M])$ contains the new sample index k). Therefore, if for all groups and all locations, there exists only one CI sequence, and there are $\sum_{i\in I^j_{q_j}} u_i = 0$ for all groups not in this CI sequence, then it is possible that for all $j \in [M]$ and $q \in [g_j]$, $\sum_{i\in I^j_q} u_i = 0$ after adding a new sample that can annihilate this CI sequence, thus Lemma 5 is violated.

In order to prove Lemma 5, it suffices to prove that for every possible group configuration groups with $\sum_{i\in I^j_q} u_i \neq 0$ exist after removing CI sequences. Sometimes only CI sequences exist, like $(u_i) - (u_i) - \cdots - (u_i)$ $(i \in [N])$, and when adding new samples, there is a risk of annihilating such CI sequences. We can avoid such situations using perturbations of network parameters (this will be explained in sections D.5.3 and D.5.4). Consequently, we give the following induction hypothesis.

For multiple locations, either non-CI sequences exist after removing each CI sequence (setting each element $\sum_{i\in I^j_{q_j}} u_i$ in it to zero), or the removal of CI sequences can be prevented.

By this induction hypothesis, after adding a new sample $\mathbf{x}_{m+1}$ into one existing group at each location, groups with $\sum_{i\in I^j_{q_j}} u_i \neq 0$ still exist, and Lemma 5 holds accordingly.

As base cases, we will prove that this induction hypothesis holds for $N = 2$. We then prove that if it holds for $N = m$ $(m \geq 2)$, it will also hold when adding a new sample $\mathbf{x}_{m+1}$, as induction step.

D.5.3 BASE CASES: $N = 2$

$M = 2$. First, we discuss the cases of two locations. When $N = 2$, all possible group configurations are discussed in the following and we will prove that induction hypothesis holds for each case.

CASE 1. Suppose the groups at two locations are

$$\begin{aligned} &\text{location 1} : (u_1, u_2)\,, \\ &\text{location 2} : (u_1, u_2)\,. \end{aligned} \tag{64}$$

If $u_1 + u_2 \neq 0$, then the sequence $(u_1, u_2) - (u_1, u_2)$ (the first element in the sequence is (u_1, u_2) from location 1, and the second one is (u_1, u_2) from location 2) is a CI sequence. However, this CI sequence cannot be vanished by adding a new sample since we can use perturbation of network parameters to prevent it from happening, described later in the induction step.

If $u_1 + u_2 = 0$, by Lemma 6, we can perturbate network parameters to obtain $u'_1 + u'_2 = 0$ at first.

CASE 2. Suppose the groups at two locations are

$$\begin{aligned} &\text{location 1} : (u_1, u_2)\,, \\ &\text{location 2} : (u_1)\,, (u_2)\,, \end{aligned} \tag{65}$$

where groups on the left have bigger v^j_is than those on the right at each location.

By Lemma 6, we can perturbate network parameters to obtain $u'_1 \neq 0$ and $u'_2 \neq 0$ for squared loss and cross-entropy loss, then the sequences $(u'_1, u'_2) - (u'_1)$ and $(u'_1, u'_2) - (u'_2)$ are both non-CI sequences, thus the induction hypothesis holds. If (u_1, u_2) in location 1 is separated as $(u'_1)\,, (u'_2)$ after perturbation, then it can be treated by the following case 3.

CASE 3. Suppose the groups at two locations are

$$\begin{aligned} &\text{location 1}: (u_1), (u_2) \\ &\text{location 2}: (u_1), (u_2). \end{aligned} \tag{66}$$

or

$$\begin{aligned} &\text{location 1}: (u_1), (u_2) \\ &\text{location 2}: (u_2), (u_1). \end{aligned} \tag{67}$$

For these cases, by Lemma 6, if $u_1 = u_2$, we can perturbate the parameters to obtain $u_1' \neq u_2'$ for squared loss and cross-entropy loss, then the sequences $(u_1') - (u_2')$ and $(u_2') - (u_1')$ are both non-CI sequences.

The sequences $(u_1) - (u_1)$ and $(u_2) - (u_2)$ are CI sequences. Again, by appropriate perturbations when adding new samples in the induction step, such CI sequences will not be removed and thus Lemma 5 holds.

$M \geq 3$. If there are more than two locations, we can select two locations as a subset, which must belong to one of the above three possible cases. The full sequences of all locations contain those sequences in this two-location subset as sub-sequences, and consequently groups with $\sum_{i \in I_{q_j}^j} u_i \neq 0$ still exist in full sequences. Therefore, induction hypothesis holds for $M \geq 3$.

D.5.4 INDUCTION STEP

At last, we prove that if the induction hypothesis holds for $N = m$ samples, then it holds for $N = m + 1$ samples. Suppose sample $\mathbf{x}_{m+1}$ is to be added given current arrangement of groups. Note that we can have $u_{m+1} \neq 0$ according to Lemma 6. There are the following three possible situations.

1) $\mathbf{x}_{m+1}$ is added as a new group at every location.

This is equivalent to adding a new CI sequence. CI and non-CI sequences formed by previous groups still exist, thus induction hypothesis holds.

2) $\mathbf{x}_{m+1}$ is inserted into an existing group at every location. More specifically, this can be divided into the following four cases.

A) $\mathbf{x}_{m+1}$ is inserted into an existing group satisfying $\sum_{i \in I_q^j} u_i = 0$ at every location.

This is equivalent to adding a new CI sequence into existing groups. Since CI or non-CI sequences still exist, the induction hypothesis holds.

B) $\mathbf{x}_{m+1}$ is inserted into existing CI or non-CI sequences.

If inserting $\mathbf{x}_{m+1}$ into an existing CI sequence, it will either be a new CI sequence, or annihilated by this addition. By induction hypothesis, non-CI sequences may still exist, hence induction hypothesis holds again after inserting $\mathbf{x}_{m+1}$. Another possible situation is that CI sequences exist, such as $(u_i) - (u_i) - \cdots - (u_i)$ $(i \in [N])$, and when adding new samples, there is a risk of annihilating such CI sequences, and if no remaining non-CI sequences exist, then no groups with $\sum_{i \in I_q^j} u_i \neq 0$ exist. We can perturbate the outputs of samples to avoid such situation using a perturbation of network parameters. For example, if $u_1 + u_3 = 0$, then adding u_3 into the sequence $(u_1) - (u_1) - \cdots - (u_1)$ will cause its disappearance. We will have $o_1 = o_3$ since sample 1 and sample 3 always appear in the same group at every location after the addition. After perturbation that tunes o_1 and o_3 differently to get $o_1' \neq o_3'$, as shown in Lemma 6, sample 1 and sample 3 then cannot appear in the same group at every location. The annihilation of sequence $(u_1) - (u_1) - \cdots - (u_1)$ is thus avoided and induction hypothesis holds.

If inserting $\mathbf{x}_{m+1}$ into an existing non-CI sequence, it will still be a non-CI sequence by definition. Therefore, induction hypothesis holds.

C) $\mathbf{x}_{m+1}$ is inserted into an existing CI sequence at some locations and into an existing non-CI sequence at remaining locations.

The involved CI sequence becomes a non-CI one, and the involved non-CI sequence may become a non-CI sequence, a CI sequence or a zero sequence. Anyway, non-CI sequences still exist and cannot be removed by adding a sample, and thus induction hypothesis holds.

D) $\mathbf{x}_{m+1}$ is inserted into an existing CI or non-CI sequence at some locations and into groups with $\sum_{i\in I_q^j} u_i = 0$ at remaining locations.

In this case, non-CI sequences will exist after inserting $\mathbf{x}_{m+1}$, thus induction hypothesis holds.

3) $\mathbf{x}_{m+1}$ is inserted into existing groups at some locations $j \in J_1$, and added as new groups at remaining locations $j \in J_2$. Equivalently, we can regard new groups in J_2 (before adding $\mathbf{x}_{m+1}$) as existing groups with $\sum_{i\in I_q^j} u_i = 0$, thus the above case D) can be applied directly, and induction hypothesis is maintained.

D.5.5 Induction Conclusion

We have proved that if the induction hypothesis holds for $N = m$ samples, then it holds for $N = m+1$ samples. By induction hypothesis, there exists $j \in [M], q \in [g_j], \ \sum_{i\in I_q^j} u_i \neq 0$, and consequently a split of data samples somewhere in feature maps is always possible.

□

D.6 Proof of Lemma 7

Proof. For cross-entropy loss, the training loss for a sample $(\mathbf{x}_i, \mathbf{y}_i)$ is

$$l\left(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i\right)) = -\sum_{j=1}^{d_y} y_{i,j} \log \frac{e^{-o_{i,j}}}{\sum_{k=1}^{d_y} e^{-o_{i,k}}}, \tag{68}$$

where $o_{i,j}$ is the jth component of $\mathbf{o}_i$. Its derivative is

$$u_i = \frac{\partial l\left(\mathbf{o}(\mathbf{x}_i), \mathbf{y}_i\right)}{\partial o_{i,1}} = \frac{e^{-o_{i,1}}}{\sum_{k=1}^{d_y} e^{-o_{i,k}}} - y_{i,1}. \tag{69}$$

Let $p_i = \frac{e^{-o_{i,1}}}{\sum_{k=1}^{d_y} e^{-o_{i,k}}}$, we have

$$u_i = p_i - y_{i,1}. \tag{70}$$

Denote $\mathbf{w}^T := W^L(1,\cdot)$, $o := o_{i,1}$. Under the perturbation $\delta\mathbf{w}$, by Taylor expansion we have

$$\begin{aligned} u_i' = u_i &+ \frac{\partial u_i}{\partial o}\left[(I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1\mathbf{x}_i)\cdot\delta\mathbf{w}\right] \\ &+ \frac{1}{2}\frac{\partial^2 u_i}{\partial o^2}\left[(I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1\mathbf{x}_i)\cdot\delta\mathbf{w}\right]^2 + O(\|\delta\mathbf{w}\|^3). \end{aligned} \tag{71}$$

Note that for $\mathbf{o}^{L-1,i} = I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1\mathbf{x}_i$, we have $\mathbf{o}^{L-1,i}(1:2P_{L-2}) = 0$ $(\forall i \in [N])$ due to the parameter setting in Lemma 1, thus perturbation $\delta\mathbf{w}(1:2P_{L-2})$ has no effect in (71).

For cross-entropy loss, the first and second order derivatives of u_i are

$$\frac{\partial u_i}{\partial o} = p_i - p_i^2, \tag{72}$$

$$\frac{\partial^2 u_i}{\partial o^2} = \frac{\partial^2 u_i}{\partial o \partial p_i}\frac{\partial p_i}{\partial o} = (1-2p_i)\frac{\partial u_i}{\partial o}, \tag{73}$$

respectively.

Let $Q^i := I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1$, we now prove that if $Q^i\mathbf{x}_i = \alpha Q^j\mathbf{x}_j$ $(\alpha \neq 0, \alpha \neq 1)$, then by setting $\delta\mathbf{w}$ appropriately, one can always make $u_i' \neq u_j'$. Note that $\frac{\partial u_i}{\partial o} = p_i - p_i^2 \neq 0$ since $p_i \neq 0$ and $p_i \neq 1$ for finite inputs. Due to the full rank assumption of $W^L I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1$ $(i \in [N])$ given in Lemma 1, we have $I^{L-1,i} \neq 0$ and consequently $Q^i\mathbf{x}_i \neq \mathbf{0},\ i \in [N]$. We can set $\delta\mathbf{w} = \epsilon\frac{Q^i\mathbf{x}_i}{\|Q^i\mathbf{x}_i\|}$ such that $(Q^i\mathbf{x}_i)\cdot\delta\mathbf{w} \neq 0$ and $(Q^j\mathbf{x}_j)\cdot\delta\mathbf{w} \neq 0$, where ϵ is a sufficiently small positive number. By (71), if $\frac{\partial u_i}{\partial o} \neq \frac{1}{\alpha}\frac{\partial u_j}{\partial o}$, then $\frac{\partial u_i}{\partial o}(Q^i\mathbf{x}_i)\cdot\delta\mathbf{w} \neq \frac{\partial u_j}{\partial o}(Q^j\mathbf{x}_j)\cdot\delta\mathbf{w}$, hence we have $u_i' \neq u_j'$ up to the first-order Taylor expansion.

If $Q^i\mathbf{x}_i = \alpha Q^j\mathbf{x}_j$ and $\frac{\partial u_i}{\partial o} = \frac{1}{\alpha}\frac{\partial u_j}{\partial o}$, we need to consider the second-order term in (71), and if $\frac{\partial^2 u_i}{\partial o^2}\alpha^2 \neq \frac{\partial^2 u_j}{\partial o^2}$, we have $u_i' \neq u_j'$ up to the second-order Taylor approximation. Otherwise, we have $\frac{\partial u_i}{\partial o} = \frac{1}{\alpha}\frac{\partial u_j}{\partial o}$ and $\frac{\partial^2 u_i}{\partial o^2}\alpha^2 = \frac{\partial^2 u_j}{\partial o^2}$, which result in

$$(p_i - p_i^2)\alpha = p_j - p_j^2 \tag{74}$$

and

$$(1 - 2p_i)\alpha = 1 - 2p_j, \tag{75}$$

respectively.

We next show that (74) and (75) cannot both hold for each case of $y_{i,1} \in \{0, 1\}$, and thus we have $u_i' \neq u_j'$ up to either the first-order or the second-order Taylor approximation.

If $u_i = p_i,\ u_j = p_j$ or $u_i = p_i - 1,\ u_j = p_j - 1$, then $u_i = u_j$ implies $p_i = p_j$. From (74) and $\alpha \neq 1$, contradiction is resulted and hence $u_i' \neq u_j'$. If samples $\mathbf{x}_i$ and $\mathbf{x}_j$ have distinct labels, say without loss of generality, $u_i = p_i - 1,\ u_j = p_j$, then $u_i = u_j$ leads to $p_j = p_i - 1$. Substituting it into (74) and (75), we obtain

$$(p_i - p_i^2)\alpha = (p_i - p_i^2) + 2(p_i - 1) \tag{76}$$

and

$$(1 - 2p_i)\alpha = (1 - 2p_i) + 2, \tag{77}$$

respectively. (76) and (77) can be transformed into

$$p_i(1 - p_i)\alpha = (1 - p_i)(p_i - 2) \tag{78}$$

and

$$(1 - p_i)(2\alpha - 2) = \alpha + 1, \tag{79}$$

respectively. By $1 - p_i \neq 0$, (78) results in

$$p_i\alpha = p_i - 2. \tag{80}$$

Solving (79) and (80), we obtain $\alpha = -1, p_i = 1$. However, $p_i = 1$ is impossible for finite inputs, therefore, we have $u_i' \neq u_j'$ after the perturbation.

□

D.7 Proof of Lemma 8

Proof. We now prove that if $u_i + u_j = 0$ and $Q^i\mathbf{x}_i = \alpha Q^j\mathbf{x}_j$ $(\alpha \neq 0, \alpha \neq 1)$, then by setting $\delta\mathbf{w}$ appropriately, one can always obtain $u_i' + u_j' \neq 0$.

By (71), if $\frac{\partial u_i}{\partial o} \neq -\frac{1}{\alpha}\frac{\partial u_j}{\partial o}$, then $\frac{\partial u_i}{\partial o}(Q^i\mathbf{x}_i)\cdot\delta\mathbf{w} \neq -\frac{\partial u_j}{\partial o}(Q^j\mathbf{x}_j)\cdot\delta\mathbf{w}$, hence $u_i' + u_j' \neq 0$ up to the first-order Taylor expansion.

If $Q^i\mathbf{x}_i = \alpha Q^j\mathbf{x}_j$ and $\frac{\partial u_i}{\partial o} = -\frac{1}{\alpha}\frac{\partial u_j}{\partial o}$, we need to consider the second-order term in (71), and if $\frac{\partial^2 u_i}{\partial o^2}\alpha^2 \neq -\frac{\partial^2 u_j}{\partial o^2}$, we have $u_i' + u_j' \neq 0$ up to the second-order Taylor approximation. Otherwise, we have $\frac{\partial u_i}{\partial o} = -\frac{1}{\alpha}\frac{\partial u_j}{\partial o}$ and $\frac{\partial^2 u_i}{\partial o^2}\alpha^2 = -\frac{\partial^2 u_j}{\partial o^2}$, which result in

$$-(p_i - p_i^2)\alpha = p_j - p_j^2 \tag{81}$$

and

$$-(1-2p_i)\alpha + (1-2p_j) = 0, \tag{82}$$

respectively.

We next show that (81) and (82) cannot both hold for every case of $y_{i,1} \in \{0,1\}$.

If $u_i = p_i,\ u_j = p_j$, then $u_i + u_j = 0$ implies $p_j = -p_i$. Substituting it into (81) and (82) and with simple calculations, we can obtain

$$-p_i\alpha - p_i = 1 - \alpha \tag{83}$$

and

$$2p_i\alpha + 2p_i = \alpha - 1. \tag{84}$$

The solution is $\alpha = 1, p_i = 0$, which contradict $\alpha \neq 1, p_i \neq 0$.

If $u_i = p_i - 1,\ u_j = p_j - 1$, then $u_i + u_j = 0$ implies $p_j = 2 - p_i$. In this case, the solution to (81) and (82) is $\alpha = 1, p_i = 1$, again contradiction is resulted.

If samples $\mathbf{x}_i$ and $\mathbf{x}_j$ have distinct labels, say without loss of generality, $u_i = p_i - 1,\ u_j = p_j$, then $u_i + u_j = 0$ leads to $p_j = 1 - p_i$. The solution to (81) and (82) will be $\alpha = -1$ and p_i can be arbitrary. However, $\alpha = -1$ implies $Q^i\mathbf{x}_i = -Q^j\mathbf{x}_j$. Due to $I^{L-1,i} \neq 0$ by the full-rank assumption, we have for all $i \in [N]$ that all components of $Q^i\mathbf{x}_i$ are nonnegative and some components of it must be positive, and thus $Q^i\mathbf{x}_i = -Q^j\mathbf{x}_j$ leads to contradiction. Therefore, $\alpha = -1$ is impossible.

In summary, for various cases of $y_{i,1} \in \{0,1\}$, we all have $u'_i + u'_j \neq 0$.

□

D.8 Proof of Lemma 6

Proof.

1. Proof of $o'_i \neq o'_j\ (i,j \in [N],\ i \neq j)$ for squared loss and cross-entropy loss.

We perturbate $W^L(1,\cdot)$ to make $o'_i \neq o'_j$. Under this perturbation, we have

$$o'_i = o_i + (I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1\mathbf{x}_i)\cdot\delta\mathbf{w} = o_i + (Q^i\mathbf{x}_i)\cdot\delta\mathbf{w}. \tag{85}$$

Since $\mathbf{o}^{L-1,i} \neq \mathbf{o}^{L-1,j}$ by Assumption 1, we have $Q^i\mathbf{x}_i \neq Q^j\mathbf{x}_j$. Then, if $o_i = o_j$, we have $o'_i - o'_j = (Q^i\mathbf{x}_i - Q^j\mathbf{x}_j)\cdot\delta\mathbf{w}$. Let $\delta\mathbf{w} = \epsilon(Q^i\mathbf{x}_i - Q^j\mathbf{x}_j)$, where ϵ is a sufficiently small positive number, there is $o'_i - o'_j = \epsilon\left\|Q^i\mathbf{x}_i - Q^j\mathbf{x}_j\right\|^2 \neq 0$. Thus, $o'_i \neq o'_j$.

2. Proof of $u'_i \neq 0\ (i \in [N])$.

For squared loss, we perturbate $W^L(1,\cdot)$ to make $u'_i \neq 0$ if $u_i = 0$. With this perturbation, we have

$$u'_i = u_i + \frac{\partial u_i}{\partial o}\left[(I^{L-1,i}W^{L-1}\cdots I^{1,i}W^1\mathbf{x}_i)\cdot\delta\mathbf{w}\right]. \tag{86}$$

We have $u_i = o_i - y_{i,1}$ for squared loss $l(\mathbf{o}_i,\mathbf{y}_i) = \frac{1}{2}\left\|\mathbf{o}_i - \mathbf{y}_i\right\|^2$, then

$$u'_i = (Q^i\mathbf{x}_i)\cdot\delta\mathbf{w}. \tag{87}$$

Setting $\delta\mathbf{w} = \epsilon(Q^i\mathbf{x}_i)$ results in $u'_i = \epsilon\left\|Q^i\mathbf{x}_i\right\|^2$. By $Q^i\mathbf{x}_i \neq \mathbf{0}$ due to the full rank requirement, we have $u'_i \neq 0$.

For cross-entropy loss function, the derivative u_i is given in (69). For finite inputs and $y_{i,1} \in \{0,1\}$, we have $\frac{e^{-o_{i,1}}}{\sum_{k=1}^{d_y} e^{-o_{i,k}}} \neq 0$ and $\frac{e^{-o_{i,1}}}{\sum_{k=1}^{d_y} e^{-o_{i,k}}} \neq 1$, thus $u_i \neq 0$.

3. Proof of $u'_i \neq u'_j$ or $u'_i + u'_j \neq 0\ (i,j \in [N],\ i \neq j)$.

For squared loss, we perturbate $W^L(1,\cdot)$ to make $u'_i \neq u'_j$ or $u'_i + u'_j \neq 0$. Under this perturbation, if $u_i = u_j$, by (86) we have $u'_i - u'_j = (Q^i\mathbf{x}_i - Q^j\mathbf{x}_j)\cdot\delta\mathbf{w}$. Let $\delta\mathbf{w} = \epsilon(Q^i\mathbf{x}_i - Q^j\mathbf{x}_j)$, there is $u'_i - u'_j = \epsilon\left\|Q^i\mathbf{x}_i - Q^j\mathbf{x}_j\right\|^2 \neq 0$. Therefore, $u'_i \neq u'_j\ (i,j \in [N],\ i \neq j)$.

If $u_i + u_j = 0$, we have $u'_i + u'_j = (Q^i\mathbf{x}_i + Q^j\mathbf{x}_j) \cdot \delta\mathbf{w}$. Note that $Q^i\mathbf{x}_i \neq -Q^j\mathbf{x}_j$ due to $I^{L-1,i} \neq 0$ and $I^{L-1,j} \neq 0$ by the full-rank assumption. Let $\delta\mathbf{w} = \epsilon(Q^i\mathbf{x}_i + Q^j\mathbf{x}_j)$, we then have $u'_i + u'_j = \epsilon \left\|Q^i\mathbf{x}_i + Q^j\mathbf{x}_j\right\|^2 \neq 0$. Therefore, $u'_i + u'_j \neq 0$ $(i, j \in [N],\ i \neq j)$.

For cross-entropy loss function, recall that $\mathbf{w}^T := W^L(1, \cdot)$ and $Q^i := I^{L-1,i}W^{L-1} \cdots I^{1,i}W^1$. Under perturbation $\delta\mathbf{w}$, by the first-order Taylor expansion in (71), using (72) we have

$$u'_i = u_i + \frac{\partial u_i}{\partial o}\left[(I^{L-1,i}W^{L-1} \cdots I^{1,i}W^1\mathbf{x}_i) \cdot \delta\mathbf{w}\right] = u_i + (p_i - p_i^2)\left[(Q^i\mathbf{x}_i) \cdot \delta\mathbf{w}\right]. \tag{88}$$

Note that $Q^i\mathbf{x}_i \neq \mathbf{0},\ Q^j\mathbf{x}_j \neq \mathbf{0}$, and $p_i - p_i^2 \neq 0$ since $p_i \neq 0$ and $p_i \neq 1$. If $Q^i\mathbf{x}_i$ is not parallel to $Q^j\mathbf{x}_j$, then $(p_i - p_i^2)(Q^i\mathbf{x}_i) \pm (p_j - p_j^2)(Q^j\mathbf{x}_j) \neq \mathbf{0}$. Therefore, we have $u'_i \neq u'_j$ or $u'_i + u'_j \neq 0$ $(i, j \in [N],\ i \neq j)$, respectively.

If $Q^i\mathbf{x}_i$ is parallel to $Q^j\mathbf{x}_j$, i.e., $Q^i\mathbf{x}_i = \alpha Q^j\mathbf{x}_j$ $(\alpha \neq 0))$, we can obtain $u'_i \neq u'_j$ or $u'_i + u'_j \neq 0$ $(i, j \in [N],\ i \neq j)$ by Lemma 7 or Lemma 8, respectively. $\alpha = 1$ is not allowed by Assumption 1.

4. $\mathbf{u} = (u_1, u_2, \cdots, u_N)^T \neq \mathbf{0}$ if $R = \frac{1}{N}\sum_{i=1}^N l(\mathbf{o}_i, \mathbf{y}_i) > 0$.

For squared loss, $u_i = o_i - y_{i,1}$. If $\mathbf{o}_i - \mathbf{y}_i = 0$ for every sample $\mathbf{x}_i$ $(i \in [N])$, then the empirical loss $R = \frac{1}{N}\sum_{i=1}^N l(\mathbf{o}_i, \mathbf{y}_i) = \frac{1}{2N}\sum_{i=1}^N \left\|\mathbf{o}_i - \mathbf{y}_i\right\|^2 = 0$, contradicting the assumption $R > 0$. Therefore, there must be some nonzero components in vector $(\mathbf{o}_i - \mathbf{y}_i)$ for some $i \in [N]$. Without loss of generality, let some components in $(o_1 - y_{1,1}, o_2 - y_{2,1}, \cdots, o_N - y_{N,1})$ be nonzero, hence $\mathbf{u} \neq \mathbf{0}$.

For cross-entropy loss, according to (69), $u_i \neq 0$ $(\forall i \in [N])$ for finite inputs, thus $\mathbf{u} = (u_1, u_2, \cdots, u_N)^T \neq \mathbf{0}$.

5. Nonzero gaps between o'_is or u'_is can still exist after subsequent perturbations, for squared loss and cross-entropy loss.

We discuss the gaps between o'_is in detail. Given some samples with identical outputs, say without loss of generality, $o_1 = o_2 = \cdots = o_s$, by Lemma 6 we can use perturbation of parameters $\delta\mathbf{w}^T := \delta W^L(1, \cdot)$ to obtain $o'_1 \neq o'_2$, thus a gap exists between o'_1 and o'_2. Without loss of generality, suppose after the first perturbation, the outputs of samples are ordered as $o'_1 > o'_2 > \cdots > o'_n = o'_{n+1} = \cdots = o'_{n+m} > o'_{n+m+1} > \cdots > o'_s$. We now want to perturbate the outputs of samples again to avoid $o'_n = o'_{n+1}$ using $\delta\mathbf{w}'$.

Under the second perturbation $\delta\mathbf{w}'$, the output of each sample is $o''_i = (\mathbf{w}' + \delta\mathbf{w}') \cdot Q^i\mathbf{x}_i := o'_i + \delta\mathbf{w}' \cdot Q^i\mathbf{x}_i$. The gaps between samples are changed to $o''_i - o''_{i+1} = o'_i - o'_{i+1} + \delta\mathbf{w}' \cdot (Q^i\mathbf{x}_i - Q^{i+1}\mathbf{x}_{i+1})$ $(i \in [s-1])$. We set $\delta\mathbf{w}'$ small enough as follows to make sure the nonzero gaps in $o'_i - o'_{i+1}$ $(i \in [s-1])$ do not disappear.

For the gap $o''_i - o''_{i+1}$, we have $o''_i - o''_{i+1} \geq o'_i - o'_{i+1} - \|\delta\mathbf{w}'\| \left\|Q^i\mathbf{x}_i - Q^{i+1}\mathbf{x}_{i+1}\right\|$ $(i \in [s-1])$. We already have $o'_i - o'_{i+1} > 0$ for nonzero gaps, and if $\delta\mathbf{w}'$ are small enough such that

$$\forall i \in [s-1],\ o'_i - o'_{i+1} - \|\delta\mathbf{w}'\| \left\|Q^i\mathbf{x}_i - Q^{i+1}\mathbf{x}_{i+1}\right\| > 0, \tag{89}$$

or equivalently

$$\forall i \in [s-1],\ \|\delta\mathbf{w}'\| < \min_{(i \in [s-1],\ o'_i - o'_{i+1} > 0)} \frac{o'_i - o'_{i+1}}{\left\|Q^i\mathbf{x}_i - Q^{i+1}\mathbf{x}_{i+1}\right\|}, \tag{90}$$

then we will have $\forall i \in [s-1],\ o''_i - o''_{i+1} > 0$. Note that $\left\|Q^i\mathbf{x}_i - Q^{i+1}\mathbf{x}_{i+1}\right\| \neq 0$. The nonzero gaps between neighboring samples still exist.

Additional perturbations can be treated in the same spirit. For a dataset with N samples, at most N perturbations are needed. The total perturbation is obtained by adding the perturbations in all steps, i.e., $\delta\mathbf{w} + \delta\mathbf{w}' + \cdots$.

The gaps between u'_is can be treated similarly using small enough perturbations.

□

D.9 PROOF OF THEOREM 3

Proof. The parameter matrix $W^1(1:P_0, 1:P_0)$ of the first convolutional layer can be implemented by a filter like $\mathbf{w}_1^1 = (0,0,1,0,0)$ (for $s_0 = 5$). With the parameter setting in Theorem 3, the output of the kth neuron in the first convolution layer is

$$o_k^1 = \sigma\left(x_i^k + c_p\right),\ \forall i \in [N] \tag{91}$$

Then, we have

$$o_k^1\left(\mathbf{x}_i\right) = \sigma\left(x_i^k + c_p\right) = x_i^k + c_p > 0,\ \forall i \in I \tag{92}$$

due to (16). The outputs of remaining neurons in the first feature map of the first convolutional layer do not matter since they will be annihilated later by W^L. Given $x_i^k + c_p < 0$ $(\forall i \notin I)$, we have

$$o_k^1\left(\mathbf{x}_i\right) = 0,\ i \in [N], i \notin I. \tag{93}$$

The filters and biases of higher convolutional layers ($l \in \{2, 3, \cdots, L-1\}$) for the first predictor propagate $o^l(1:P_{l-1})$ (the outputs in the first feature map of each layer) forward without changing them.

Sufficiently large positive constants c_l $(l \in [L-1])$ are added to the biases of each layer such that the ReLU units in the second subnetwork are all activated and their outputs are not truncated. The first convolutional layer will be added c_1 to each of its output neuron, and the change of outputs of higher convolutional layers can be recursively computed as

$$\begin{aligned}
&\Delta o^1(i) = c_1,\ \ i \in [P_0+1:n_1],\\
&\Delta o^l(i) = \sum_{j\in[P_{l-2}+1:n_{l-1}]} W^l(i,j)\Delta o^{l-1}(j) + c_l,\ i \in [P_{l-1}+1:n_l],\ l \in \{2,3,\cdots,L-1\},\\
&\Delta o^L(i) = \sum_{j\in[P_{L-2}+1:n_{L-1}]} W^L(i,j)\Delta o^{L-1}(j),\ i \in [d_y].
\end{aligned} \tag{94}$$

The effect of introducing c_l $(l \in [L-1])$ is cancelled at the output layer by subtracting $\Delta o^L(i)$ from $o^L(i)$ $(i \in [d_y])$. This is equivalent to subtracting $\Delta o^L(i)$ from b_i^L.

By the parameter setting, the output of each sample is

$$\mathbf{o}\left(\mathbf{x}_i\right) = W_2\hat{\mathbf{x}}_i + \sigma\left(x_i^k + c_p\right)\cdot\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T,\ \ \forall i \in [N], \tag{95}$$

where the expression of $W_2\hat{\mathbf{x}}_i$ is given in (25). This implies

$$\mathbf{o}\left(\mathbf{x}_i\right) = W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T,\ \ i \in I, \tag{96}$$

and

$$\mathbf{o}\left(\mathbf{x}_i\right) = \tilde{\mathbf{y}}_i,\quad l\left(\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) = 0,\ \ i \in I \tag{97}$$

if there is a single index in set I, where $\tilde{\mathbf{y}}_i$ is defined in section 4. If there are multiple elements in set I,

$$\sum_{i\in I} l\left(\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) = \sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T, \mathbf{y}_i\right) \tag{98}$$

is minimized.

For samples $\{\mathbf{x}_i,\ i \notin I\}$, we have

$$\mathbf{o}\left(\mathbf{x}_i\right) = W_2\hat{\mathbf{x}}_i, \forall i \in [N],\ i \notin I. \tag{99}$$

We now show that for the CNN given in Theorem 3, $\theta = \left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is a local minimum. This can be seen by perturbating the parameters and showing the non-decreaseness of loss. The training loss is

$$\begin{aligned}
R &= \frac{1}{N}\sum_{I=1}^{N} l\left(\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right)\\
&= \frac{1}{N}\sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T, \mathbf{y}_i\right) + \frac{1}{N}\sum_{\substack{i\in[N],\\ i\notin I}} l\left(W_2\hat{\mathbf{x}}_i, \mathbf{y}_i\right).
\end{aligned} \tag{100}$$

Under perturbation $\left(\delta W^l, \delta \mathbf{b}^l\right)_{l=1}^L$, since there is a gap between x_i^k $(i \in [N])$ and the threshold $\frac{1}{2}\left(x_{i_*}^k + x_{j_*}^k\right)$, the ReLU unit in (91) keeps its activation status for every sample when the perturbation of $b_1^1 = c_p$ is sufficiently small. Therefore, the output of each sample is still in the form of (95). Accordingly, if there are multiple elements in set I,

$$\begin{aligned} R' :=& \frac{1}{N}\sum_{i=1}^{N} l\left(\mathbf{o}\left(\mathbf{x}_i\right) + \delta\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) \\ =& \frac{1}{N}\sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T + \delta\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) \\ &+ \frac{1}{N}\sum_{\substack{i\in[N]\\ i\notin I}} l\left(W_2\hat{\mathbf{x}}_i + \delta\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right), \end{aligned}$$

where $\delta\mathbf{o}\left(\mathbf{x}_i\right)$ $(i \in I)$ and $\delta\mathbf{o}\left(\mathbf{x}_i\right)$ $(i \notin I)$ are caused by the perturbations of parameters in both W_1 and W_2. However, since $\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)$ minimize $\sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T, \mathbf{y}_i\right)$, and the space of $\mathbf{o}\left(\mathbf{x}_i\right)$ has been fully explored by $\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)$ during minimization, we have

$$\begin{aligned} &\sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T + \delta\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) \\ &\geq \sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T, \mathbf{y}_i\right). \end{aligned} \tag{101}$$

On the other hand, since the size of each convolutional layer satisfies $n_l - P_{l-1} \geq d_y$ $(l \in [L-1])$, the matrix $\hat{W}^L(\cdot, P_{L-2}+1 : n_{L-1})\hat{W}^{L-1}(P_{L-2}+1 : n_{L-1}, P_{L-3}+1 : n_{L-2}) \cdots \hat{W}^1(P_0+1 : n_1, \cdot) \in \mathbb{R}^{d_y \times d_x}$ is full rank and can generate the full space of $d_y \times d_x$ matrices around θ, thus $\sum_{\substack{i\in[N],\\ i\notin I}} l\left(W_2\hat{\mathbf{x}}_i, \mathbf{y}_i\right)$ is minimized by W_2 in (25). Therefore,

$$\sum_{\substack{i\in[N],\\ i\notin I}} l\left(W_2\hat{\mathbf{x}}_i + \delta\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) \geq \sum_{\substack{i\in[N],\\ i\notin I}} l\left(W_2\hat{\mathbf{x}}_i, \mathbf{y}_i\right). \tag{102}$$

Using (101) and (102),

$$\begin{aligned} R' &\geq \frac{1}{N}\sum_{i\in I} l\left(W_2\hat{\mathbf{x}}_i + \left(x_i^k + c_p\right)\left(\lambda_1, \lambda_2, \cdots, \lambda_{d_y}\right)^T, \mathbf{y}_i\right) + \frac{1}{N}\sum_{\substack{i\in[N],\\ i\notin I}} l\left(W_2\hat{\mathbf{x}}_i, \mathbf{y}_i\right) \\ &= \frac{1}{N}\sum_{I=1}^{N} l\left(\mathbf{o}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) = R. \end{aligned} \tag{103}$$

For the case of single element in set I, $R' \geq R$ can be shown easily due to (97). Therefore, $\theta = \left(W^l, \mathbf{b}^l\right)_{l=1}^L$ is a local minimum.

□