Journal of Machine Learning Research 26 (2025) 1-31 Submitted 9/24; Revised 5/25; Published 6/25

Interpretable Global Minima of Deep ReLU Neural Networks on Sequentially Separable Data

Thomas Chen TC@MATH.UTEXAS.EDU
Department of Mathematics
University of Texas at Austin
Austin, TX 78712 USA

Patrícia Muñoz Ewald* EWALD@UTEXAS.EDU
Department of Mathematics
University of Texas at Austin
Austin, TX 78712 USA

Editor: Christoph Lampert

Abstract

We explicitly construct zero loss neural network classifiers. We write the weight matrices and bias vectors in terms of cumulative parameters, which determine truncation maps acting recursively on input space. The configurations for the training data considered are (i) sufficiently small, well separated clusters corresponding to each class, and (ii) equivalence classes which are sequentially linearly separable. In the best case, for Q classes of data in $\mathbb{R}^M$, global minimizers can be described with $Q(M+2)$ parameters.

Keywords: deep learning, classification, loss landscape, interpolation, interpretability, geometry

1. Introduction

Deep learning algorithms are often thought to involve a trade-off between accuracy and interpretability. Highly overparameterized neural networks are capable of interpolating generic data, and gradient descent offers a standard way of training the parameters to minimize some loss function. However, the resulting models function as black boxes, providing limited insight into their internal mechanisms. In contrast, underparameterized networks cannot achieve zero training loss unless the data possesses sufficient structure (Chen and Ewald, 2025; Chen and Moore, 2025).

In this work, continuing the program started by the authors (Chen and Ewald, 2023a,b), we provide an interpretation for the action of the layers of a neural network, and explicitly construct global minima for a classification task with well distributed data, in a manner that is independent of the number of training samples. This characterization is done in terms of a map which encodes the action of a layer given by a weight matrix W, a bias vector b, and activation function σ,

$$\tau_{W,b}(x) = (W)^{+} \left(\sigma(Wx + b) - b\right),$$

*. Corresponding author.

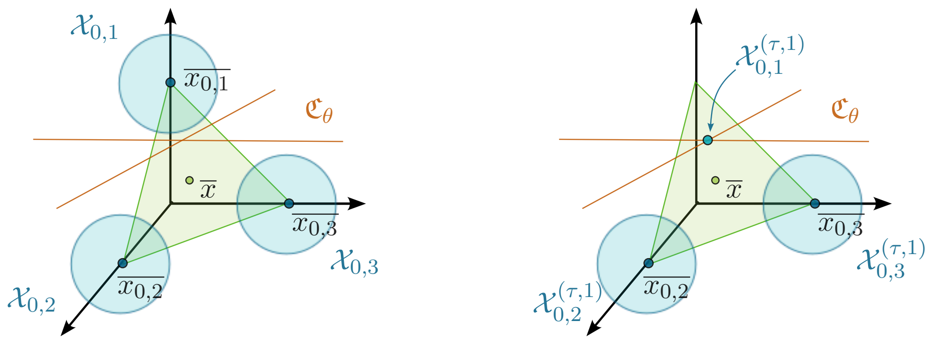

Figure 1: A schematic representation of the action of a layer given by a truncation map. On the left, the original data with three classes contained in balls around the means $\overline{x_{0,j}}$, with $\mathcal{X}_{0,1}$ sitting in the backward cone $\mathfrak{C}_\theta$, and the remaining classes sitting in the forward cone. On the right, the truncation map has pushed all of $\mathcal{X}_{0,1}$ to a single point, the base of the cone. In green, we see the 2-simplex generated by the means, and the barycenter $\overline{x}$. The geometry of the data corresponds to the assumptions of Proposition 3.

where $(W)^+$ denotes the generalized inverse of W. A neural network with L layers can be seen as an application of $L-1$ such transformations τ on input space, which curate the data, followed by an affine function in the last layer mapping to the output space,

$$x^{(L)} = W^{(L)}(x)^{(\tau,L-1)} + b^{(L)},$$

where $x^{(\tau,L-1)}$ represents the $L-1$ consecutive applications of τ maps to an input x, and $W^{(L)}, b^{(L)}$ are appropriate cumulative parameters. While this works for any given activation function, we specialize to the case of a ReLU neural network, and in this context call τ the *truncation map*. An incomplete, but highly intuitive visualization of its action comes from looking at cones: For any cone in input space, there exist W and b such that $\tau_{W,b}$ acts as the identity on the forward cone, and projects the entirety of the backward cone to the base point (see Lemma 2).

Consider data in $\mathbb{R}^M$ to be classified into Q classes. In previous work (Chen and Ewald, 2023a), the tools outlined above were used to construct explicit global minimizers for a network with constant width $M = Q$ and "clustered" data. In this paper, we extend this to the case where the width of the hidden layers is allowed to decrease monotonically from M to Q, for $Q \leq M$. As a corollary, we show a reduction in the total number of parameters needed to interpolate the data. Our first result (informally stated) is

Theorem (Corollary to Proposition 3)**.** *Consider a set of training data $\mathcal{X}_0 = \bigcup_{j=1}^{Q} \mathcal{X}_{0,j} \subset \mathbb{R}^M$ separated into Q classes corresponding to linearly independent labels in $\mathbb{R}^Q$, for $Q \leq M$. If the data is distributed in sufficiently small and separated balls ("clustered"), then there exists a ReLU neural network with $Q(M+Q^2)$ parameters in $Q+1$ layers which interpolates the data.*

The weights and biases can be explicitly and geometrically described. Moreover, this does not depend on the size of the data set, $|\mathcal{X}_0|$.

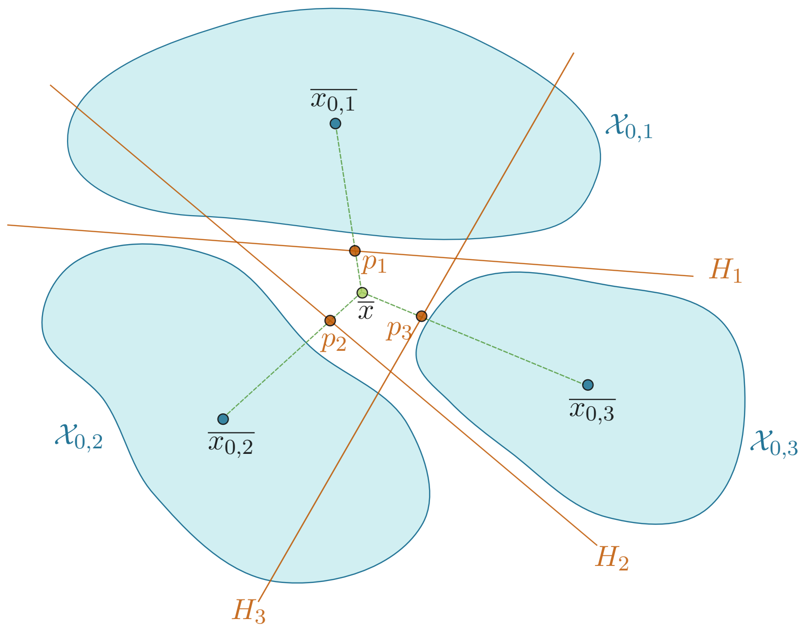

Figure 2: A 2-dimensional representation of a data set $\mathcal{X}_0 = \bigcup_{j=1}^{3} \mathcal{X}_{0,j}$ that is sequentially linearly separable (Definition 5), but not clustered (cf. Figure 1). Note that the ordering chosen is the only possible one. The geometry of the data corresponds to the assumptions of Theorem 6.

The proof can be easily sketched: For each layer, a truncation map can be constructed which maps an entire cluster to a new point, while keeping the remaining clusters the same (see Figure 1). The weights and biases can be given in terms of the variables defining the associated cones.

As the angular aperture of a cone gets larger, it becomes a hyperplane. These are a well known classification tool, and there are algorithms dedicated to finding hyperplanes maximizing the distance to the two sets being separated, such as support vector machines (SVMs). Recent works show that under some circumstances, ReLU networks trained with gradient descent to perform binary classification tend to find max-margin hyperplanes (see e.g., Phuong and Lampert, 2020).[1]

The same algorithms can be adapted to the case of multiple classes, by reducing to binary classification via several one-versus-all or one-versus-one strategies. However, a neural network tasked with performing multiclass classification must yield a single function capable of classifying. When the data is clustered as above, Figure 1 shows how the layers of the network will push each cluster into a single point. By considering hyperplanes as limits of cones, this can be generalized to data for which at each step, one class gets mapped into a single point. Thus, each layer acts to find a one-versus-all hyperplane. If this can be done so that at the end there are Q linearly independent points, the last layer will complete the

1. The tendency of gradient-based algorithms to find critical points which generalize well or are simple in some way, e.g. low-rank, is called implicit bias or implicit regularization.

classification with zero loss. We define data which satisfies this condition as *sequentially linearly separable*, and this leads to our next result (informally stated).

Theorem (Theorem 6)**.** *Consider a set of training data $\mathcal{X}_0 = \bigcup_{j=1}^{Q} \mathcal{X}_{0,j} \subset \mathbb{R}^M$ separated into Q classes corresponding to linearly independent labels in $\mathbb{R}^Q$. If the data is sequentially linearly separable, then a ReLU neural network with $Q+1$ layers of size $d_0 = \cdots = d_Q = M \geq Q$, $d_{Q+1} = Q$, attains a degenerate global minimum with zero training cost, which can be parametrized by*

$$\{(\theta_\ell, \nu_\ell, \mu_\ell)\}_{\ell=1}^{Q} \subset (0, \pi) \times \mathbb{R}^M \times (0, 1),$$

corresponding to triples of angles, normal vectors and a line segment of base points (resp.) describing cones and hyperplanes. The parameters do not depend on the size of the data set, $|\mathcal{X}_0|$.

In the work at hand, we focus on geometric structure in zero-loss neural networks, and leave questions about algorithmic implementations to future work. We note that the complexity of algorithmic implementations will address the detection of the training data geometry, instead of the determination of network parameters. Once the training data geometry is known, the network parameters can be calculated non-algorithmically.

In the case $M = Q$, the inductive proof of the main result was adapted to give $2^Q - 1$ suboptimal local minima, associated to the binary choice for each cluster to have the truncation map act trivially or collapse the data to a point (Chen and Ewald, 2023a). However, the proof of minimality relied on an exact expression for the norm of local minima which holds in this case (Chen and Ewald, 2023b, Theorem 3.2). The assumption $M = Q$ is dropped for our results here, so the local minima cannot be found in the same way. Moreover, the sequential separability of the data in Theorem 6 might depend on the ordering of the classes (cf. Figure 2), so that the truncation maps cannot be chosen independently.

Finally, we note that while we do not consider layers with arbitrary width, increasing the width would only add unnecessary parameters, as we assume the data to be suitably linearly separable. It is known that data which is not linearly separable can be classified by an SVM with the use of kernel machines (Boser et al., 1992) which map the data into a space of higher dimension in such a way that, in this new space, a separating hyperplane can be found. Moreover, neural networks approximate kernel machines in the infinite width limit (Lee et al., 2018; Jacot et al., 2018). With these considerations, it could be the case that for data that is not linearly separable, the first layers of a neural network could be seen as describing a map to a higher dimensional space where the conditions for sequential linear separability are now satisfied, and the results above can be applied. An implementation of this idea for binary classification is given by Ewald (2025).

1.1 Related Work

We give here a short and incomplete survey of related results.

Since the ReLU function acts as the identity when activated, it is instructive to compare with the linear case. The **loss landscape of linear neural networks** is very well understood. Its study started with the classic work by Baldi and Hornik (1989), which proved (under some assumptions on the architecture and data) that a shallow (one hidden layer) linear neural network with square loss has only global minima and saddle points, and that every critical point is a projection of the least-squares estimator onto a subspace spanned by the eigenvectors of a relevant matrix determined by the data and label matrices.[2] Kawaguchi (2016) extended this to deep linear networks, proving under milder conditions (both on data and architecture) that there are no suboptimal local minima, and giving some properties of the Hessian of the loss at saddle points. Zhou and Liang (2018) provided complete characterizations of all critical points of a deep linear network with square loss, arbitrary architecture and very mild assumptions on the data. Recently, Achour et al. (2024) gave a complete second order analysis of the loss landscape of deep linear networks, with the same assumptions as Kawaguchi (2016).

The **loss landscape of ReLU neural networks**, on the other hand, is not as well understood. Following assumptions adopted in work by Choromanska et al. (2015), Kawaguchi (2016) treats the activation functions as random variables, and what is computed is the loss with expectation; under these assumptions, the loss reduces to the loss of a deep linear network. Tian (2017) studied ReLU networks with no hidden layers, Gaussian distributed input data and target function a neural network with fixed weights, and found that certain types of critical points are not isolated. Zhou and Liang (2018) studied shallow ReLU networks with very mild assumptions on the data, characterized all critical points of the square loss for parameters in certain regions of parameter space (depending on the activation pattern of the neurons, and not covering the whole space), and gave an example of a network for which there is a suboptimal local minimum. While not studying critical points of a loss function directly, Grigsby et al. (2022, 2023) studied functional properties of ReLU neural networks, such as "functional dimension"[3], the level sets of which are contained in level sets of any loss function.

Finally, the results here show how a trained neural network could take within-class variability to zero. This is one of the features of **neural collapse**, a phenomenon which has been empirically shown by Papyan et al. (2020); Han et al. (2022) to emerge when training a neural network towards zero loss.

1.2 Outline of Paper

In Section 2, we clearly define the types of neural networks and the cost function under consideration and fix notation. In Section 3, we define the truncation map, remark on its action in terms of cones in input space and cumulative parameters, and prove key lemmas. In Section 4, we give precise statements for Proposition 3 and a corollary. In Section 5, we define sequentially linearly separable data, and present a precise statement and proof for Theorem 6. In Section 6.1, we briefly discuss generalization guarantees, and in Section 6.2, we elaborate on the relationship between our results and neural collapse. In Section 7, we prove Proposition 3. Finally, we collect some basic results on geometry of cones in Appendix A.

2. That is, a critical point with $rank(W_2W_1) = r$ satisfies $W_2W_1 = U_rU_r^TYX^+$, where U_r is formed by r columns of U, the matrix of eigenvectors of $\Sigma = Y(X)^+XY^T$. See next section for notation.
3. A simplified definition of functional dimension is the rank of the Jacobian of the network with respect to its parameters.

2. Setting and Notation

We will be dealing with neural networks of the form

$$\begin{aligned} x^{(0)} &= x_0 \in \mathbb{R}^{d_0} \quad \text{initial input}, \\ x^{(\ell)} &= \sigma(W_\ell x^{(\ell-1)} + b_\ell) \in \mathbb{R}^{d_\ell}, \text{ for } \ell = 1, \cdots, L-1, \\ x^{(L)} &= W_L x^{(L-1)} + b_L \in \mathbb{R}^{d_L}, \end{aligned} \tag{1}$$

where $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ for $\ell = 1, \cdots, L$ are weight matrices, $b_\ell \in \mathbb{R}^{d_\ell}$ for $\ell = 1, \cdots, L$ are bias vectors, and the activation function σ is the ramp function which acts component-wise on vectors, $\sigma(x)_i = (x_i)_+ = \max(x_i, 0)$. When considering the action on a data matrix X_0 with N columns corresponding to N data points, we write

$$X^{(\ell)} = \sigma\left(W_\ell X^{(\ell-1)} + B_\ell\right),$$

where

$$B_\ell := b_\ell u_N^T \in \mathbb{R}^{d_\ell \times N}$$

for

$$u_N := (1, \cdots, 1)^T \in \mathbb{R}^N.$$

Consider a set of training data $\mathcal{X}_0 = \bigcup_{i=1}^{Q} \mathcal{X}_{0,j} \subset \mathbb{R}^M$, where points in each subset $\mathcal{X}_{0,j} = \{x_{0,j,i}\}_{i=1}^{N_j}$ are associated to one of Q linearly independent labels, $y_j \in \mathbb{R}^Q$. Let $\sum_{j=1}^{Q} N_j = N$, so that $|\mathcal{X}_0| = N$. We will be interested in the average of training inputs associated to each output y_j,

$$\overline{x_{0,j}} := \frac{1}{N_j} \sum_{i=1}^{N_j} x_{0,j,i},$$

and the differences

$$\Delta x_{0,j,i} := x_{0,j,i} - \overline{x_{0,j}}.$$

It will be convenient to arrange the data and related information into matrices, and so we make the following definitions,

$$\begin{aligned} X_{0,j} &:= \left[x_{0,j,1} \cdots x_{0,j,i} \cdots x_{0,j,N_j}\right] && \in \mathbb{R}^{M \times N_j}, \\ X_0 &:= \left[X_{0,1} \cdots X_{0,j} \cdots X_{0,Q}\right] && \in \mathbb{R}^{M \times N}, \\ \Delta X_{0,j} &:= \left[\Delta x_{0,j,1} \cdots \Delta x_{0,j,i} \cdots \Delta x_{0,j,N_j}\right] && \in \mathbb{R}^{M \times N_j}, \\ \Delta X_0 &:= \left[\Delta X_{0,1} \cdots \Delta X_{0,j} \cdots \Delta X_{0,Q}\right] && \in \mathbb{R}^{M \times N}, \\ \overline{X_{0,j}} &:= X_{0,j} - \Delta X_{0,j} && \in \mathbb{R}^{M \times N_j}, \\ \overline{X_0} &:= \left[\overline{X_{0,1}} \cdots \overline{X_{0,j}} \cdots \overline{X_{0,Q}}\right] && \in \mathbb{R}^{M \times N}, \\ \overline{X_0^{red}} &:= \left[\overline{x_{0,1}} \cdots \overline{x_{0,j}} \cdots \overline{x_{0,Q}}\right] && \in \mathbb{R}^{M \times Q}, \\ Y &:= \left[y_1 \cdots y_j \cdots y_Q\right] && \in \mathbb{R}^{Q \times Q}, \\ Y^{ext} &:= \left[y_1 u_{N_1}^T \cdots y_j u_{N_j}^T \cdots y_Q u_{N_Q}^T\right] && \in \mathbb{R}^{Q \times N}. \end{aligned} \tag{2}$$

Then $\mathcal{X}_{0,j} = \mathrm{Col}(X_{0,j})$, the set of columns of $X_{0,j}$, and similarly $\mathcal{X}_0 = \mathrm{Col}(X_0)$.

Throughout, we will assume that $\overline{X_0^{red}}$ and Y have full rank $Q \leq M$, i.e., that the averages $\overline{x_{0,j}}$ and output vectors y_j are linearly independent. Since $\overline{X_0^{red}}$ is injective, we write its generalized inverse as

$$(\overline{X_0^{red}})^+ = (\overline{X_0^{red}}^T \overline{X_0^{red}})^{-1} \overline{X_0^{red}}^T \in \mathbb{R}^{Q\times M},$$

and this is a left inverse of $\overline{X_0^{red}}$. In general, for any matrix $A \in \mathbb{R}^{m\times n}$, we let $(A)^+ \in \mathbb{R}^{n\times m}$ denote its generalized inverse, that is, the unique matrix satisfying

$$\begin{gathered} A(A)^+A = A, \\ (A)^+A(A)^+ = (A)^+, \\ A(A)^+ \text{ and } (A)^+A \text{ are symmetric.} \end{gathered} \tag{3}$$

Recall that $(A)^+A$ is an orthogonal projector to $(\ker(A))^\perp$, $A(A)^+$ is an orthogonal projector to $\mathrm{ran}\,(A)$, and $(A)^+$ is a left (resp. right) inverse of A if it is injective (resp. surjective).

Finally, for an appropriate neural network defined as in (1) with $d_0 = M$ and $d_L = Q$, we consider the square loss, given by the $\mathcal{L}^2$ Schatten (or Hilbert-Schmidt) norm,

$$\|A\|_{\mathcal{L}^2} := \sqrt{\mathrm{Tr}(AA^T)},$$

so that the cost function is

$$\begin{aligned} \mathcal{C}[X_0, Y^{ext}, (W_i, b_i)_{i=1}^L] &:= \frac{1}{2}\left\| X^{(L)} - Y^{ext} \right\|_{\mathcal{L}^2}^2 \\ &= \frac{1}{2}\sum_{j=1}^{Q}\sum_{i=1}^{N_j}\left| x_{0,j,i}^{(L)} - y_j \right|^2, \end{aligned}$$

where $|\cdot|$ denotes the usual euclidean norm. When the data set given by X_0 and Y is clear by context, we will simply denote

$$\mathcal{C}[(W_i, b_i)_{i=1}^L] := \mathcal{C}[X_0, Y^{ext}, (W_i, b_i)_{i=1}^L].$$

3. Truncation Map, Cones and Cumulative Parameters

We recall here the definition of the truncation map (Chen and Ewald, 2023a), making the necessary modifications for the case where the width is not constant and the weight matrices might not be square.

Definition 1. Given $W \in \mathbb{R}^{M_1\times M_0}$ and $b \in \mathbb{R}^{M_1}$ we define the truncation map

$$\begin{aligned} \tau_{W,b} : \mathbb{R}^{M_0} &\to \mathbb{R}^{M_0} \\ x &\mapsto (W)^+ \left(\sigma(Wx + b) - b\right), \end{aligned}$$

where $(W)^+$ is the generalized inverse defined as in (3).

Similarly, we can define the action on a data matrix $X \in \mathbb{R}^{M_0 \times N}$

$$\tau_{W,b}(X) = (W)^{+}\left(\sigma(WX + B) - B\right),$$

where $B = b\, u_N^T$. This map encodes the action of a layer of a neural network and represents it in the domain. It has a nice recursive property which makes it possible to extend this representation to several layers. The following is a small generalization of a previous result (Chen and Ewald, 2023a) to the case where the dimension of the intermediate spaces given by the hidden layers is allowed to decrease.

Proposition 2. *Assume $M = d_0 \geq d_1 \geq \cdots \geq d_L = Q$, $X^{(\ell)} \in \mathbb{R}^{d_\ell \times N}$ corresponds to the output of a hidden layer of a neural network defined as in (1) on a data matrix $X_0 \in \mathbb{R}^{M \times N}$, and all of the associated weight matrices $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ are full rank. Then the truncation map defined satisfies*

$$X^{(\ell)} = W_\ell\, \tau_{W_\ell, b_\ell}(X^{(\ell-1)}) + B_\ell. \tag{4}$$

Moreover, defining the cumulative parameters

$$W^{(\ell)} := W_\ell \cdots W_1 \in \mathbb{R}^{d_\ell \times d_0}, \quad \text{for } \ell = 1, \cdots, L, \tag{5}$$

and

$$b^{(\ell)} := \begin{cases} W_\ell \cdots W_2 b_1 + W_\ell \cdots W_3 b_2 + \cdots + W_\ell b_{\ell-1} + b_\ell, & \text{if } \ell \geq 2, \\ b_1, & \text{if } \ell = 1, \end{cases} \tag{6}$$

we have

$$X^{(\ell)} = W^{(\ell)} (X_0)^{(\tau,\ell)} + B^{(\ell)}, \tag{7}$$

for $\ell = 1, \cdots, L-1$, and

$$X^{(L)} = W^{(L)} (X_0)^{(\tau,L-1)} + B^{(L)}, \tag{8}$$

where

$$\begin{aligned} (X_0)^{(\tau,\ell)} &:= \tau_{W^{(\ell)}, b^{(\ell)}}(\tau_{W^{(\ell-1)}, b^{(\ell-1)}}(\cdots \tau_{W^{(2)}, b^{(2)}}(\tau_{W^{(1)}, b^{(1)}}(X^{(0)})) \cdots) \\ &= \tau_{W^{(\ell)}, b^{(\ell)}}(X_0)^{(\tau,\ell-1)}. \end{aligned}$$

Proof Using the fact that $W_\ell (W_\ell)^{+} = \mathbf{1}_{d_\ell \times d_\ell}$ for all $\ell = 1, \cdots, L$, we have

$$\begin{aligned} X^{(\ell)} &= \sigma(W_\ell X^{(\ell-1)} + B_\ell) \\ &= W_\ell (W_\ell)^{+} \sigma(W_\ell X^{(\ell-1)} + B_\ell) \\ &= W_\ell (W_\ell)^{+} \left(\sigma(W_\ell X^{(\ell-1)} + B_\ell) - B_\ell\right) + B_\ell \\ &= W_\ell\, \tau_{W_\ell, B_\ell}(X^{(\ell-1)}) + B_\ell, \end{aligned}$$

which proves (4). Going one step further,

$$\begin{aligned}
X^{(\ell)} &= \sigma(W_\ell X^{(\ell-1)} + B_\ell) \\
&= \sigma(W_\ell (W_{\ell-1}\, \tau_{W_{\ell-1},b_{\ell-1}}(X^{(\ell-2)}) + B_{\ell-1}) + B_\ell) \\
&\overset{(*)}{=} W_\ell W_{\ell-1} (W_\ell W_{\ell-1})^+ \sigma(W_\ell (W_{\ell-1}\, \tau_{W_{\ell-1},b_{\ell-1}}(X^{(\ell-2)}) + B_{\ell-1}) + B_\ell) \\
&= W_\ell W_{\ell-1} (W_\ell W_{\ell-1})^+ \sigma(W_\ell W_{\ell-1}\, \tau_{W_{\ell-1},b_{\ell-1}}(X^{(\ell-2)}) + W_\ell B_{\ell-1} + B_\ell) \\
&= W_\ell W_{\ell-1}\, \tau_{W_\ell W_{\ell-1}, W_\ell b_{\ell-1} + b_\ell}(\tau_{W_{\ell-1},b_{\ell-1}}(X^{(\ell-2)})) + W_\ell B_{\ell-1} + B_\ell,
\end{aligned}$$

where $(*)$ follows from the surjectivity of $W_\ell W_{\ell-1}$. An iteration of these arguments then yields (7). Finally,

$$\begin{aligned}
X^{(L)} &= W_L X^{(L-1)} + B_L \\
&= W_L (W^{(L-1)} X^{(\tau,L-1)} + B^{(L-1)}) + B_L \\
&= W^{(L)} X^{(\tau,L-1)} + B^{(L)}.
\end{aligned}$$

■

This lemma, and especially expression (8), suggest that we think of a neural network with L layers as a concatenation of $L-1$ truncation maps acting on the data in input space $\mathbb{R}^{d_0}$, followed by an affine map to the appropriate output space $\mathbb{R}^{d_L}$. These truncation maps and the affine map are given by cumulative parameters, so that from now on in this paper a layer will be defined by prescribing $(W^{(\ell)}, b^{(\ell)})$ instead of (W_ℓ, b_ℓ), for all $\ell = 1 \cdots, L$.

We will need the following properties of the cumulative parameters in the case where $d_0 \geq \cdots \geq d_L$, which applies to the results in Sections 4 and 5.

Lemma 1. Consider a collection $\{W_\ell\}_{\ell=1}^L$ of linear maps which can be composed into surjective cumulative matrices $\{W^{(\ell)}\}_{\ell=1}^L$, defined as in (5). Define

$$\begin{aligned}
P^{(\ell)} &:= (W^{(\ell)})^+ W^{(\ell)}\, (W^{(\ell-1)})^+ W^{(\ell-1)} \cdots (W^{(1)})^+ W^{(1)} \\
&= (W^{(\ell)})^+ W^{(\ell)} P^{(\ell-1)}.
\end{aligned} \tag{9}$$

Then

$$P^{(\ell)} = (W^{(\ell)})^+ W^{(\ell)}, \tag{10}$$

and

$$W^{(\ell)} P^{(\ell-1)} = W^{(\ell)}. \tag{11}$$

Proof Note that $P^{(1)} = (W^{(1)})^+ W^{(1)}$ by definition, and assume $P^{(\ell-1)} = (W^{(\ell-1)})^+ W^{(\ell-1)}$. If $W^{(\ell-1)}$ is surjective, then $W^{(\ell-1)} (W^{(\ell-1)})^+ = \mathbf{1}$ and

$$\begin{aligned}
P^{(\ell)} &= (W^{(\ell)})^+ W^{(\ell)} P^{(\ell-1)} \\
&= (W^{(\ell)})^+ W^{(\ell)} (W^{(\ell-1)})^+ W^{(\ell-1)} \\
&= (W^{(\ell)})^+ W_\ell \left(W^{(\ell-1)} (W^{(\ell-1)})^+ \right) W^{(\ell-1)} \\
&= (W^{(\ell)})^+ W_\ell W^{(\ell-1)} = (W^{(\ell)})^+ W^{(\ell)},
\end{aligned}$$

which gives (10).

We have proved that $P^{(\ell)}$ is the projector onto the orthogonal complement of the kernel of $W^{(\ell)}$ for all $\ell = 1, \cdots, L$. Now, let $Q^{(\ell-1)} = \mathbf{1} - P^{(\ell-1)}$ be the projector to the kernel of $W^{(\ell-1)}$. Then

$$\begin{aligned} W^{(\ell)}(\mathbf{1} - P^{(\ell-1)}) &= W^{(\ell)} Q^{(\ell-1)} \\ &= W_\ell W^{(\ell-1)} Q^{(\ell-1)} \\ &= 0, \end{aligned}$$

and we have proved (11). ■

The next lemma is the key tool used to prove the main results in Sections 4 and 5, as it describes the action of the truncation map in some particular cases. The activation function σ acts in such a way that it is the identity on the positive sector

$$\mathbb{R}^n_+ := \left\{ (x_1, \cdots, x_n)^T \in \mathbb{R}^n : x_i \geq 0, i = 1, \cdots, n \right\},$$

and the zero map on the negative sector $\mathbb{R}^n_-$ (similarly defined). Figure 3 shows the action of the truncation map in two dimensions, for $W \in \mathbb{R}^{2\times 2}$ and $b \in \mathbb{R}^2$. In this simple context, it is clear that for an invertible weight matrix, the action of $\tau_{W,b}$ is completely determined by a cone (given by the blue dashed lines on the first picture). While this is not always true, it is still fruitful to consider a similar picture in general.

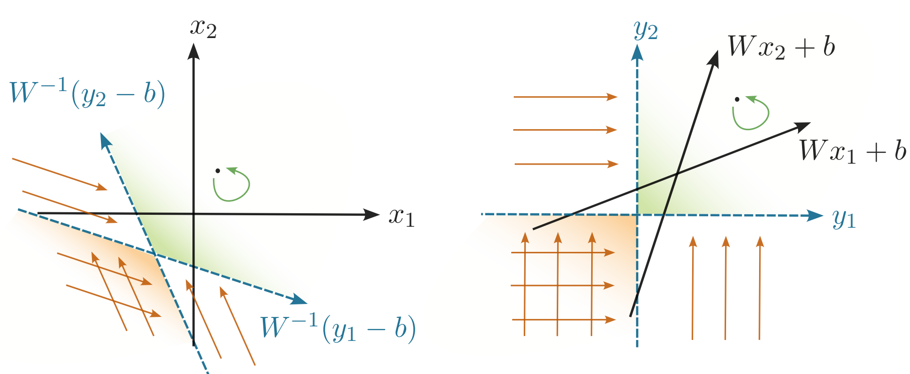

Figure 3: The action of $\tau_{W,b}$ in two dimensions. The image on the left represents the input space given by coordinates (x_1, x_2). On the right we see the image of this space under the map $x \mapsto \sigma(Wx + b)$, where σ acts according to the new coordinate system (y_1, y_2): it is the identity on the positive sector (shaded in green), and can be thought of as projecting along the orange arrows in the other regions, which gives the total effect of mapping the negative sector (shaded in orange) to the origin. Pulling this action back to input space produces the cone on the picture to the left.

We define a cone of angle $\theta \le \pi$ around a given vector $h \in \mathbb{R}^n$ and based at the origin,

$$\mathfrak{C}_\theta[h] := \left\{ x \in \mathbb{R}^n : \angle(x,h) \le \frac{\theta}{2} \right\}. \tag{12}$$

When the forward direction h is clear, we will refer to a pair $\mathfrak{C}_\theta[h]$, $\mathfrak{C}_\theta[-h]$ as the forward and backward cones, respectively. Then, for

$$\theta_n := 2 \arccos \frac{\sqrt{n-1}}{\sqrt{n}}, \tag{13}$$

we note that $\mathfrak{C}_{\theta_n}[u_n] \subset \mathbb{R}^n$ is the largest cone centered around the diagonal $u_n = (1, \cdots, 1)^T$ that is contained fully in $\mathbb{R}^n_+$.[4] Similarly, the backward cone is fully contained in $\mathbb{R}^n_-$. A cone $\mathfrak{C}_\theta[u_n]$ of larger angle can be made to fit into $\mathbb{R}^n_+$ by a transformation which fixes the axis u_n and shrinks every other direction perpendicular to the axis according to

$$\lambda(\theta, \theta_n) := \begin{cases} \frac{\tan \frac{\theta_n}{2}}{\tan \frac{\theta}{2}} < 1, & \text{if } \theta > \theta_n, \\ 1, & \text{otherwise.} \end{cases} \tag{14}$$

Now, in the case where there is no change in dimension, Lemma 2 below states that for any given cone we can construct W and b so that the truncation map $\tau_{W,b}$ acts as the identity on the forward cone, and projects the entirety of the backward cone to its base point. However, if $\tau_{W,b}$ is encoding a layer in a network for which there is a reduction in dimension, its image must also undergo a dimensional reduction. To account for the possible decrease in width of the network, consider the family of projectors

$$\begin{aligned} &\mathcal{I}(n,m) : \mathbb{R}^n \to \mathbb{R}^m, \quad n \ge m, \\ &\mathcal{I}(n,m)(e_i^n) = \begin{cases} e_i^m, & i = 1, \cdots, m, \\ 0, & \text{else,} \end{cases} \end{aligned} \tag{15}$$

where $\{e_i^n\}_{i=1}^n$ is the standard basis for $\mathbb{R}^n$, so that the matrix $\mathcal{I}(n,m) \in \mathbb{R}^{m \times n}$ is a block matrix consisting of an identity matrix $\mathbf{1}_{m\times m}$ and zeroes.

Lemma 2. Consider the cone $p + \mathfrak{C}_\theta[h] \subset \mathbb{R}^n$ centered around a unit vector $h \in \mathbb{R}^n$ and based at $p \in \mathbb{R}^n$, and consider $W : \mathbb{R}^n \to \mathbb{R}^m$, $b \in \mathbb{R}^m$ such that

$$\begin{aligned} W &= \mathcal{I}(n,m) W_\theta R, \\ b &= -Wp, \end{aligned}$$

where

$$W_\theta = \tilde{R} \operatorname{diag}(1, \lambda(\theta,\theta_n), \cdots, \lambda(\theta,\theta_n))\, \tilde{R}^T \in \mathbb{R}^{n\times n}, \tag{16}$$

with $\lambda(\theta, \theta_n)$ defined as in (14) and $R, \tilde{R} \in O(n, \mathbb{R})$ are such that

$$Rh = \frac{u_n}{|u_n|}, \ \tilde{R} e_1^n = \frac{u_n}{|u_n|}.$$

4. See appendix.

Then

$$\tau_{W,b}(x) = \begin{cases} ((W)^{+}W)\,x, & x \in p + \mathfrak{C}_\theta[h], \\ ((W)^{+}W)\,p, & x \in p + \mathfrak{C}_\theta[-h]. \end{cases}$$

Moreover, if $P : \mathbb{R}^n \to \mathbb{R}^n$ is a map such that $WP = W$, then

$$\tau_{W,b}(Px) = \tau_{W,b}(x). \tag{17}$$

Proof First, consider $x \in p + \mathfrak{C}_\theta[h]$. Since

$$\tau_{W,b}(x) = (W)^{+}\left(\sigma(Wx + b) - b\right),$$

it suffices to show that $Wx + b \in \mathbb{R}^n_+$, and so $\sigma(Wx + b) = Wx + b$. Substituting the given expressions for W and b,

$$Wx + b = W(x - p) = \mathcal{I}(n, m)W_\theta R(x - p),$$

then $x \in p + \mathfrak{C}_\theta[h]$ implies $(x - p) \in \mathfrak{C}_\theta[h]$, and we observe the action of the maps on the cones. First, an orthogonal transformation changes the axis, but not the opening angle, of a cone, and we have

$$R(x - p) \in \mathfrak{C}_\theta[Rh] = \mathfrak{C}_\theta[u_n].$$

Then, the map W_θ is constructed so that it fixes the axis, but changes the aperture of the cone so that $W_\theta\mathfrak{C}_\theta[u_n] \subset \mathfrak{C}_{\theta_n}[u_n]$, which is contained entirely in $\mathbb{R}^n_+$, so

$$W_\theta R(\mathfrak{C}_\theta[h]) \subset \mathfrak{C}_{\theta_n}[u_n] \subset \mathbb{R}^n_+,$$

and $W(x - p) \in \mathbb{R}^m_+$.

For the backward cone, $x \in p + \mathfrak{C}_\theta[-h]$, similar arguments yield

$$W(\mathfrak{C}_\theta[-h]) \subset \mathfrak{C}_{\theta_n}[-u_n] \subset \mathbb{R}^n_-,$$

and then $W(x - p) \in \mathbb{R}^m_-$ implies $\sigma(Wx + b) = 0$ and

$$\tau_{W,b}(x) = (W)^{+}(-b) = (W)^{+}Wp.$$

Finally, (17) follows easily from the definition of the truncation map,

$$\begin{aligned} \tau_{W,b}(Px) &= (W)^{+}\left(\sigma(W(Px) + b) - b\right) \\ &= (W)^{+}\left(\sigma(Wx + b) - b\right) \\ &= \tau_{W,b}(x). \end{aligned}$$

■

We can now use the truncation map and Lemma 2 to generalize the construction of zero loss global minima (Chen and Ewald, 2023a) in two different directions. First, in Section 4, we keep the training data as sufficiently separated, small enough clusters, but allow for $M = d_0 \geq d_1 \geq \cdots \geq d_L = Q$. Then, in Section 5, we hold the width of every layer but the last one constant, $d_0 = d_1 = \cdots = d_{L-1}$, but allow for $M > Q$ and for a more general distribution of the training data.

Remark 3. Note that the decomposition

$$W = W_\theta R$$

coincides with the polar decomposition of a matrix

$$W = |W|\, R$$

for $|W| := (WW^T)^{\frac{1}{2}}$ and $R \in O(n, \mathbb{R})$. Indeed,

$$\begin{aligned} WW^T &= \tilde{R}\,\mathrm{diag}(1, \cdots)\tilde{R}^T R R^T \tilde{R}\,\mathrm{diag}(1, \cdots)\tilde{R}^T \\ &= \tilde{R}\,\mathrm{diag}(1, \cdots)^2 \tilde{R}^T \\ &= W_\theta^2. \end{aligned}$$

4. Clustered Data and Dimensional Reduction

We will consider in this section the same geometry on the data as that adopted by Chen and Ewald (2023a), which we now briefly recall. Let

$$\overline{x} := \frac{1}{Q}\sum_{j=1}^{Q} \overline{x_{0,j}}$$

be the average of all class means, and note that this is not necessarily the average of all data points. We will assume that

$$\delta := \sup_{i,j} |\Delta x_{0,j,i}| < c_0 \min_j |\overline{x_{0,j}} - \overline{x}|\,, \tag{18}$$

for $c_0 \in (0, \frac{1}{4})$. Moreover, consider

$$f_j := \frac{\overline{x} - \overline{x_{0,j}}}{|\overline{x} - \overline{x_{0,j}}|}, \quad j = 1, \cdots, Q, \tag{19}$$

the unit vectors pointing from $\overline{x_{0,j}}$ to $\overline{x}$. For each $j = 1, \cdots, Q$, let $\theta_{*,j}$ be the smallest angle such that

$$\bigcup_{j' \neq j} B_{4\delta}(\overline{x_{0,j'}}) \subset \overline{x_{0,j}} + \mathfrak{C}_{\theta_{*,j}}[f_j],$$

and assume

$$\max_j \theta_{*,j} < \pi. \tag{20}$$

Proposition 3. *Consider a set of training data $\mathcal{X}_0 = \bigcup_{j=1}^{Q} \mathcal{X}_{0,j} \subset \mathbb{R}^M$ separated into Q classes corresponding to linearly independent labels $\{y_j\}_{j=1}^{Q} \subset \mathbb{R}^Q$. If conditions* (18) *and* (20) *are satisfied, then a neural network with ReLU activation function defined as in* (1)*, with $L = Q + 1$ layers, $d_0 = M$, $d_{Q+1} = Q$, and $d_{\ell-1} \geq d_\ell \geq Q$ for all $\ell = 1, \cdots, L$, attains*

$$\min_{(W_i, b_i)_{i=1}^{L}} \mathcal{C}[(W_i, b_i)_{i=1}^{L}] = 0,$$

with weights and biases given recursively by (5), (6) *and the following cumulative parameters: for* $\ell = 1, \cdots, Q$,

$$\begin{aligned} W^{(\ell)} &= \mathcal{I}(d_0, d_\ell) W_{\theta_*} R_\ell, \\ b^{(\ell)} &= -W^{(\ell)}(\overline{x_{0,\ell}} + \mu_\ell f_\ell), \end{aligned}$$

for $\mathcal{I}(d_0, d_\ell) \in \mathbb{R}^{d_\ell \times d_0}$ *a projection,* $W_{\theta_*} \in GL(d_0, \mathbb{R})$ *and* $R_\ell \in O(d_0, \mathbb{R})$ *constructed as in Section 3, and some* $\mu_\ell \in (2\delta, 3\delta)$*; for* $\ell = Q + 1$,

$$\begin{aligned} W^{(Q+1)} &= Y(\overline{X_0^{red}}^{(\tau,Q)})^+, \\ b^{(Q+1)} &= 0. \end{aligned}$$

This minimum is degenerate.

Since the only constraints on the architecture are that the widths of the layers must be non-increasing and bounded below by Q, one could construct a network for which the first hidden layer immediately reduces to Q dimensions, and the following layers have constant width. This leads to a reduction in the total number of parameters needed, and we have proved

Corollary 4. *For training data* $\mathcal{X}_0 = \bigcup_{j=1}^{Q} \mathcal{X}_{0,j} \subset \mathbb{R}^M$ *satisfying the same conditions as above, there exists a neural network with* $Q(M + Q^2)$ *parameters interpolating the data, and this does not depend on* $|\mathcal{X}_0| = N$.

Thus, well-distributed data makes it possible to attain zero loss even in the under-parametrized setting.

We defer the full proof to Section 7.

5. Hyperplanes and Sequentially Separable Data

In the previous section, we assumed that the data had to be neatly contained in sufficiently small and well spaced balls. However, when using Lemma 2 in the proof, it was sufficient to check that each ball lay inside a certain backward or forward cone. More generally, for the action of any given layer described by a truncation map, all that matters is that there exists a cone such that the class that is being singled out sits entirely within the backward cone, and the remaining data sits in the forward cone.

From this point of view, the action of each iteration of the truncation map can be thought of as performing a one-versus-all classification: a particular class is being singled out and separated from the remaining classes. However, what we obtain is different from simply breaking down a multiclass classification problem into several binary classification tasks. Data that has been truncated by a layer will remain truncated, and so the order in which the one-versus-all tasks are done is important. To account for this, we extend the notion of linear separability from binary classification to a problem with multiple classes as follows.

Definition 5. We say a data set is *sequentially linearly separable* if there exists an ordering of the classes such that for each $j = 1, \cdots, Q$, there exists a hyperplane H_j and a point $p_j \in H_j$ such that H_j separates $\mathcal{X}_{0,j}$ and $\bigcup_{k<j}\{p_k\} \cup \bigcup_{k>j} \mathcal{X}_{0,k}$.[5]

5. See Figure 2 in the introduction for an example.

Let $H(p,\nu)$ be the hyperplane defined by a point p and a normal vector ν,

$$H(p,\nu) := \{p + x \in \mathbb{R}^n : \langle x, \nu\rangle = 0\}.$$

For data that is sequentially linearly separable, we have the corresponding hyperplanes $H_j = H(p_j, h_j)$ for some unit vectors h_j, $j = 1, \cdots, Q$. We will always consider data sets to be finite, and so the distance from any such hyperplane to the each of the two sets it is separating is always positive. Moreover, a pair $(p,\nu) \subset \mathbb{R}^n \times \mathbb{R}^n$, along with an angle $\theta < \pi$, define a pair of cones $p + \mathfrak{C}_\theta[\nu]$ and $p + \mathfrak{C}_\theta[-\nu]$ as in (12), and so to each hyperplane $H(p,\nu)$ is associated a family of cones.

Remark 4. Similarly, we note that there exists $\epsilon > 0$ such that each point p_j can be freely perturbed on an ϵ-ball around it (possibly moving the corresponding hyperplane H_j as well) so that $(\tilde{p}_j, \tilde{H}_j)_{j=1}^Q$ still satisfies Definition 5, for $\tilde{p}_j \in (p_j + B_\epsilon(p_j)) \cap \tilde{H}_j$, $j = 1, \cdots, Q$. Therefore, we may assume that $\{p_j\}_{j=1}^Q$ form a linearly independent set.

Theorem 6. *Consider a set of training data $\mathcal{X}_0 = \bigcup_{j=1}^Q \mathcal{X}_{0,j} \subset \mathbb{R}^M$ separated into Q classes corresponding to linearly independent labels $\{y_j\}_{j=1}^Q \subset \mathbb{R}^Q$. If the data is sequentially linearly separable, then a neural network with ReLU activation function defined as in* (1)*, with $L = Q + 1$ layers, $d_0 = M$, $d_{Q+1} = Q$, and $d_0 = d_\ell \geq Q$ for all hidden layers $\ell = 1, \cdots, Q$, attains*

$$\min_{(W_i,b_i)_{i=1}^L} \mathcal{C}[(W_i, b_i)_{i=1}^L] = 0,$$

with weights and biases given recursively by (5), (6) *for the following cumulative parameters: for $\ell = 1, \cdots, Q$,*

$$\begin{aligned} W^{(\ell)} &= W_{\theta_\ell} R_\ell, \\ b^{(\ell)} &= -W^{(\ell)} p_\ell, \end{aligned}$$

for $W_{\theta_\ell} \in GL(d_0, \mathbb{R})$ and $R_\ell \in O(d_0, \mathbb{R})$ constructed as in Section 3; and for $\ell = Q + 1$,

$$\begin{aligned} W^{(Q+1)} &= Y(\overline{X_0^{red}}^{(\tau,Q)})^+, \\ b^{(Q+1)} &= 0. \end{aligned}$$

This minimum is degenerate.

Proof For $j = 1, \cdots, Q$, let

$$H_j = H(p_j, h_j) \tag{21}$$

be a sequence of hyperplanes that realizes the sequential linear separability of the given data, and assume the ordering is correct as given. As $\mathcal{X}_0$ is a finite set, for each j there exists $\theta_{j,min} < \pi$ such that any $\theta_j \in (\theta_{j,min}, \pi)$ implies

$$\begin{aligned} p_j + \mathfrak{C}_{\theta_j}[h_j] &\supset \bigcup_{k<j} \{p_k\} \cup \bigcup_{k>j} \mathcal{X}_{0,k} \\ &\text{and} \\ p_j + \mathfrak{C}_{\theta_j}[-h_j] &\supset \mathcal{X}_{0,j}. \end{aligned} \tag{22}$$

We will prove that $X_{0,j}^{(\tau,Q)} = p_j u_{N_j}^T$, or that for all $x \in \mathcal{X}_{0,j}$, $x^{(\tau,Q)} = p_j$, for $j = 1, \cdots, Q$.

Induction base $\ell = 1$. Given p_1, h_1, and $\theta_1 \in (\theta_{1,min}, \pi)$, we have

$$\begin{aligned} \mathcal{X}_{0,1} &\subset p_1 + \mathfrak{C}_{\theta_1}[-h_1], \\ \bigcup_{j>1} \mathcal{X}_{0,j} &\subset p_1 + \mathfrak{C}_{\theta_1}[h_1]. \end{aligned} \tag{23}$$

Let

$$W_1 = W^{(1)} := W_{\theta_1} R_1 \in \mathbb{R}^{d_0 \times d_0},$$

for W_{θ_1} defined as in (16), and $R_1 \in O(d_0, \mathbb{R})$ such that

$$R_1 h_1 = \frac{u_{d_0}}{|u_{d_0}|},$$

and

$$b_1 = b^{(1)} := -W_1 p_1.$$

Then Lemma 2 implies we have the following truncation map

$$\tau_{W^{(1)},b^{(1)}}(x) = \begin{cases} x, & x \in p_1 + \mathfrak{C}_{\theta_1}[h_1], \\ p_1, & x \in p_1 + \mathfrak{C}_{\theta_1}[-h_1], \end{cases}$$

which, according to (23), acts on the data as

$$X_{0,j}^{(\tau,1)} = \tau_{W^{(1)},b^{(1)}}(X_{0,j}) = \begin{cases} p_1 u_{N_1}^T, & j = 1, \\ X_{0,j}, & j > 1, \end{cases}$$

so that the first layer has successfully shrunk the first equivalence class of data points $\mathcal{X}_{0,1}$ to a single point.

The induction step $\ell - 1 \to \ell$. Assume

$$X_{0,j}^{(\tau,\,\ell-1)} = \begin{cases} p_j u_{N_j}^T, & j \le \ell - 1, \\ X_{0,j}, & j > \ell - 1, \end{cases} \tag{24}$$

where p_j are the points given by (21). We will construct cumulative parameters $W^{(\ell)}, b^{(\ell)}$ such that

$$X_{0,j}^{(\tau,\ell)} = \tau_{W^{(\ell)},b^{(\ell)}}(X_{0,j}^{(\tau,\ell-1)}) = \begin{cases} p_j u_{N_j}^T = X_{0,j}^{(\tau,\ell-1)}, & j < \ell, \\ p_\ell u_{N_\ell}^T, & j = \ell, \\ X_{0,j} = X_{0,j}^{(\tau,\ell-1)}, & j > \ell. \end{cases} \tag{25}$$

By sequential linear separability of the data, there exists a hyperplane $H_\ell = H(p_\ell, h_\ell)$ in the collection given by (21) separating $\mathcal{X}_{0,\ell}$ and $\bigcup_{j<\ell}\{p_j\} \cup \bigcup_{j>\ell} \mathcal{X}_{0,j}$. We can use the induction hypothesis (24) to rewrite

$$\mathcal{X}_{0,\ell} = \mathcal{X}_{0,\ell}^{(\tau,\ell-1)} \text{ and } \bigcup_{j<\ell}\{p_j\} \cup \bigcup_{j>\ell} \mathcal{X}_{0,j} = \bigcup_{j\neq\ell} \mathcal{X}_{0,j}^{(\tau,\ell-1)},$$

and so H_ℓ separates $\mathcal{X}_{0,\ell}^{(\tau,\ell-1)}$ and $\bigcup_{j\neq\ell}\mathcal{X}_{0,j}^{(\tau,\ell-1)}$.

Then by (22), there is a $\theta_\ell < \pi$ such that

$$\bigcup_{j\neq\ell}\mathcal{X}_{0,j}^{(\tau,\ell-1)} \subset p_\ell + \mathfrak{C}_{\theta_\ell}[h_\ell] \quad \text{and} \quad \mathcal{X}_{0,\ell} \subset p_\ell + \mathfrak{C}_{\theta_\ell}[-h_\ell]. \tag{26}$$

Now we can choose

$$\begin{aligned} W^{(\ell)} &:= W_{\theta_\ell} R_\ell \in \mathbb{R}^{d_0\times d_0}, \\ b^{(\ell)} &:= -W^{(\ell)}\, p_\ell \in \mathbb{R}^{d_0}, \end{aligned} \tag{27}$$

for W_{θ_ℓ} defined as in (16) and $R_\ell \in O(d_0,\mathbb{R})$ such that

$$R_\ell h_\ell = \frac{u_{d_0}}{|u_{d_0}|},$$

so that Lemma 2 gives

$$\tau_{W^{(\ell)},b^{(\ell)}}(x) = \begin{cases} x, & x \in p_\ell + \mathfrak{C}_{\theta_\ell}[h_\ell], \\ p_\ell, & x \in p_\ell + \mathfrak{C}_{\theta_\ell}[-h_\ell], \end{cases}$$

which, along with (26), implies (25).

Conclusion of proof. At the end of Q layers, or Q applications of the truncation maps $\tau_{W^{(\ell)},b^{(\ell)}}$, we have mapped each equivalence class $\mathcal{X}_{0,j}$ to a single point, $(\mathcal{X}_{0,j})^{(\tau,Q)} = p_j$, for all $j = 1, \cdots, Q$. This implies that $(\Delta X_0)^{(\tau,Q)} = 0$.

In the last layer, we have

$$X^{(Q+1)} = W^{(Q+1)}(X_0)^{(\tau,Q)} + B^{(Q+1)},$$

and we wish to minimize [6]

$$\min_{W^{(Q+1)},b^{(Q+1)}} \left\| W^{(Q+1)}(X_0)^{(\tau,Q)} + B^{(Q+1)} - Y^{ext} \right\|_{\mathcal{L}^2}^2 .$$

This can be achieved by letting

$$b^{(Q+1)} := 0,$$

and solving

$$W^{(Q+1)}(X_0)^{(\tau,Q)} = Y^{ext}.$$

Since

$$\begin{aligned} (X_0)^{(\tau,Q)} &= (\overline{X_0} + \Delta X_0)^{(\tau,Q)} \\ &= (\overline{X_0})^{(\tau,Q)} + (\Delta X_0)^{(\tau,Q)} \\ &= (\overline{X_0})^{(\tau,Q)}, \end{aligned}$$

6. Recall the matrices defined in (2).

we can equivalently solve

$$W^{(Q+1)}(\overline{X_0^{red}})^{(\tau,Q)} = Y. \tag{28}$$

As the collection $\{p_j\}_{j=1}^Q$ is assumed linearly independent (see Remark 4),

$$(\overline{X_0^{red}})^{(\tau,Q)} = [p_1 \cdots p_Q]$$

satisfies

$$((\overline{X_0^{red}})^{(\tau,Q)})^+(\overline{X_0^{red}})^{(\tau,Q)} = \mathbf{1}_{Q\times Q},$$

and so

$$W^{(Q+1)} = Y((\overline{X_0^{red}})^{(\tau,Q)})^+ \tag{29}$$

is a solution to (28), and we have in fact produced a collection of weights and biases $(W_i^{**}, b_i^{**})_{i=1}^{Q+1}$ given by the cumulative parameters defined in (27) and (29) that achieves zero cost,

$$\mathcal{C}[(W_i^{**}, b_i^{**})_{i=1}^{Q+1}] = 0.$$

Finally, the minimum we have constructed is degenerate: for a fixed collection of hyperplanes, the angles $(\theta_\ell)_{\ell=1}^Q$ of the associated cones can vary in $(\theta_{1,min}, \pi) \times \cdots \times (\theta_{Q,min}, \pi)$, and so parametrize a family of cumulative weights $(W^{(\ell)}[\theta_\ell])_{\ell=1}^Q$. ■

6. Additional Remarks

We make here a few remarks concerning generalization and neural collapse.

6.1 Generalization

Let $\rho \in \mathbb{P}(Z)$ be a probability distribution on $Z = X \times Y$, the set of data $x \in X$ and labels $y \in Y \subset \mathbb{R}$. Following the most general setup in the book by Vapnik (2013), we consider

$$\{Q(z,\alpha), \alpha \in \Lambda\}$$

to be a parametrized set of functions representing the composition of a loss function with a function $f_\alpha : X \to Y$. The real risk of a function f_α on data distributed according to ρ is

$$R(\alpha) := \int_Z Q(z,\alpha)\,\mathrm{d}\rho(z).$$

Given a training set $\{z_1, \cdots, z_n\}$, the empirical risk is

$$R_{emp}(\alpha) := \frac{1}{n}\sum_{i=1}^n Q(z_i,\alpha).$$

Then if $a \leq Q(z, \alpha) \leq b$, the following holds for any $\alpha \in \Lambda$ and with probability at least $1 - \eta$:

$$R(\alpha) \leq R_{emp}(\alpha) + (b - a)\sqrt{\frac{VC(\Lambda)\log(n+1) - \log(\frac{\eta}{4})}{n}},$$

where $VC(\Lambda)$ is the Vapnik–Chervonenkis dimension of the function class parametrized by Λ. In particular, for an empirical risk minimizer which achieves zero loss,

$$R(\alpha_*) \leq (b - a)\sqrt{\frac{VC(\Lambda)\log(n+1) - \log(\frac{\eta}{4})}{n}}. \tag{30}$$

In this work, $X = \mathbb{R}^{d_0}$, $Y = \mathbb{R}^Q$, the functions $f_\alpha : \mathbb{R}^{d_0} \to \mathbb{R}^Q$ are ReLU neural networks parametrized by weights and biases, and we consider the squared loss, so

$$Q(z = (x, y), \alpha) = |f_\alpha(x) - y|^2 .$$

The loss function can be bounded, for instance, by imposing bounds on the norms of the parameters and the input data. See work by Chen et al. (2025) for a more detailed discussion.

If the VC dimension of the function class is finite,[7] then (30) implies that the results obtained in the work at hand generalize well to test data which is identically distributed to the training data, as they do not depend on the size of the training data set.

6.2 Cost Decomposition and Neural Collapse

Note that since the constructions in this paper do not depend on the cost function chosen, we can change it to average over each class separately,[8]

$$\begin{aligned}\mathcal{C}_{\mathcal{N}}[(W_i, b_i)_{i=1}^L] &:= \sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left| x_{0,j,i}^{(L)} - y_j \right|^2 \\ &= \left\| X^{(L)} - Y^{ext} \right\|_{\mathcal{L}^2_{\mathcal{N}}}^2 ,\end{aligned}$$

7. It is known that the VC dimension of real valued ReLU neural networks is finite, and explicit bounds exist (see e.g., Bartlett et al., 2019).
8. Equivalently, we could consider balanced classes.

where $\mathcal{N} := \mathrm{diag}(N_j \mathbf{1}_{N_j \times N_j} | j = 1, \cdots, Q)$ and $\langle A, B \rangle_{\mathcal{L}^2_{\mathcal{N}}} := \mathrm{Tr}(A\mathcal{N}^{-1}B^T)$. For simplicity, throughout this section we will refer to $\mathcal{C}_{\mathcal{N}}$ as $\mathcal{C}$. This can be decomposed as follows

$$\begin{aligned}
\sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left| x_{0,j,i}^{(L)} - y_j \right|^2 &= \sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left(\left| x_{0,j,i}^{(L)} \right|^2 - 2\langle x_{0,j,i}^{(L)}, y_j \rangle + |y_j|^2 \right) \\
&= \sum_{j=1}^{Q} \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \left| x_{0,j,i}^{(L)} \right|^2 - \left| \overline{x_{0,j}^{(L)}} \right|^2 + \left| \overline{x_{0,j}^{(L)}} \right|^2 - 2\langle \frac{1}{N_j} \sum_{i=1}^{N_j} x_{0,j,i}^{(L)}, y_j \rangle + \frac{N_j}{N_j} |y_j|^2 \right) \\
&= \sum_{j=1}^{Q} \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \left| x_{0,j,i}^{(L)} - \overline{x_{0,j}^{(L)}} \right|^2 + \left| \overline{x_{0,j}^{(L)}} - y_j \right|^2 \right) \qquad (31) \\
&= \sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left| \Delta x_{0,j,i}^{(L)} \right|^2 + \sum_{j=1}^{Q} \left| \overline{x_{0,j}^{(L)}} - y_j \right|^2 \\
&= \sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left| W^{(L)} \Delta x_{0,j,i}^{(\tau,L-1)} \right|^2 + \sum_{j=1}^{Q} \left| W^{(L)} \overline{x_{0,j}^{(\tau,L-1)}} + b^{(L)} - y_j \right|^2 .
\end{aligned}$$

Since both terms are non-negative, the cost will be zero if, and only if, each term is zero, and so we have the following interpretation for the results presented in this paper: the truncation maps implemented by the hidden layers have the role of minimizing the first term, by which all variances are reduced to zero, while the last (affine) layer minimizes the second term, whereby the class averages are matched with the reference outputs y_j.

Han, Papyan, and Donoho (2022) observed that when training a classifier to zero mean square loss, the following phenomena are observed on the penultimate layer features:

- (NC1) Within-class variability collapse, which for instance occurs if $\Delta X_0^{(L-1)} \to 0$.
- (NC2) Convergence to simplex ETF: the vectors $(\overline{x_{0,j}^{(L-1)}} - \overline{x^{(L-1)}})$ evolve to have maximal angle between them, and the same length, forming an equilateral simplex, or simplex equiangular tight frame (ETF).
- (NC3) Convergence to self-duality: $f_j^{(L-1)} + w_j \to 0$, where $f_j^{(L-1)}$ are unit vectors pointing from $\overline{x_{0,j}^{(L-1)}}$ to $\overline{x^{(L-1)}}$, defined analogously to (19), and w_j are the rows of W_L, for $j = 1, \cdots, Q$.

They consider balanced classes and decompose the mean square loss into the sum of a least-squares term and the remainder, and then further decompose the least-squares term into terms associated to the properties above:

$$\mathcal{C} = \mathcal{C}_{LS} + \mathcal{C}_{LS}^{\perp} = (\mathcal{C}_{NC1} + \mathcal{C}_{NC2,3}) + \mathcal{C}_{LS}^{\perp}. \qquad (32)$$

Further, they provide empirical evidence that $\mathcal{C}_{LS}^{\perp}$ becomes negligible fast during training.

We will make some remarks comparing (32) with the cost decomposition (31) and the results in this paper. The last line of (31) can be written in two different ways,

$$\mathcal{C} = \sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left| W^{(L)} \Delta x_{0,j,i}^{(\tau,L-1)} \right|^2 + \sum_{j=1}^{Q} \left| W^{(L)} \overline{x_{0,j}^{(\tau,L-1)}} + b^{(L)} - y_j \right|^2 \tag{33}$$

$$= \sum_{j=1}^{Q} \frac{1}{N_j} \sum_{i=1}^{N_j} \left| W_L \Delta x_{0,j,i}^{(L-1)} \right|^2 + \sum_{j=1}^{Q} \left| W_L \overline{x_{0,j}^{(L-1)}} + b_L - y_j \right|^2 . \tag{34}$$

If we let $b_L = 0$ and $W_L = W_{LS}$ be the least squares matrix that minimizes the second term in (34), then $\mathcal{C} = \mathcal{C}_{LS}$, the first term in (34) is $\mathcal{C}_{NC1}$, and the second term is $\mathcal{C}_{NC2,3}$ (up to constants). Comparing with the interpretation given after (31), this suggests that the truncation maps in the hidden layers drive NC1, while the last-layer affine map is responsible for NC2,3.

Let us compare the last-layer weights and biases obtained by minimizing according to (34) and (33). For the rest of this section we take $Y = \mathbf{1}_{Q\times Q}$, $\overline{X^{(\tau,\ell)}} := \left[\overline{x_{0,1}^{(\tau,\ell)}} \cdots \overline{x_{0,Q}^{(\tau,\ell)}} \right]$ and $\overline{X^{(\ell)}} := \left[\overline{x_{0,1}^{(\ell)}} \cdots \overline{x_{0,Q}^{(\ell)}} \right]$. In this work we have minimized with respect to (33),

$$\tilde{W}^{(L)} = (\overline{X^{(\tau,L-1)}})^{+},\ b^{(L)} = 0,$$

which yields

$$\begin{aligned} \tilde{W}_L &= \tilde{W}^{(L)} (W^{(L-1)})^{+} \\ &= (\overline{X^{(\tau,L-1)}})^{+} (W^{(L-1)})^{+}. \end{aligned}$$

Minimizing with respect to (34), as Han et al. (2022), gives

$$W_L = (\overline{X^{(L-1)}})^{+},\ b_L = 0.$$

We rewrite W_L: from Proposition 2,

$$\begin{aligned} \overline{X^{(L-1)}} &= W^{(L-1)} \overline{X^{(\tau,L-1)}} + B^{(L-1)} \\ &= W^{(L-1)} \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right) \end{aligned}$$

for

$$\beta^{(L-1)} := (W^{(L-1)})^{+} B^{(L-1)},$$

assuming $W^{(L-1)}$ is surjective. Define the orthogonal projectors

$$\mathcal{P}_W := \mathcal{P}_{(\ker W^{(L-1)})^{\perp}} = (W^{(L-1)})^{+} W^{(L-1)}, \tag{35}$$

$$\mathcal{P}_X := \mathcal{P}_{\operatorname{ran}(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)})} = \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right) \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right)^{+} . \tag{36}$$

Then

$$\begin{aligned} W_L &= \left(W^{(L-1)} \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right) \right)^+ \\ &\overset{(1)}{=} \left(\mathcal{P}_W \left[\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right] \right)^+ \left(W^{(L-1)} \mathcal{P}_X \right)^+ , \end{aligned}$$

where (1) follows from a result that can be found in the book by Campbell and Meyer (2009, Theorem 1.4.1). Now, observe that

$$\operatorname{ran} \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right) \subset \operatorname{ran} (W^{(L-1)})^+ = \left(\ker W^{(L-1)} \right)^{\perp}, \tag{37}$$

so

$$\mathcal{P}_W \mathcal{P}_X = \mathcal{P}_X \mathcal{P}_W = \mathcal{P}_X,$$

and we conclude

$$W_L = \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right)^+ \left(W^{(L-1)} \mathcal{P}_X \right)^+ .$$

If the sets in (37) are equal, then $\mathcal{P}_W = \mathcal{P}_X$ and W_L further simplifies to

$$W_L = \left(\overline{X^{(\tau,L-1)}} + \beta^{(L-1)} \right)^+ \left(W^{(L-1)} \right)^+ ,$$

so W_L and $\tilde{W}_L$ would differ only in the $\beta^{(L-1)}$ term. This could be achieved, for instance, by taking $d_{L-1} = Q$ and $\overline{X^{(\tau,L-1)}} \in \mathbb{R}^{M \times Q}$ injective (recall we are already assuming $W^{(L-1)}$ to be surjective). Suppose this is the case, for simplicity, and consider two subcases:

- First, if $b^{(L-1)} = 0$, then

  $$W_L = \tilde{W}_L \text{ and } b^{(L)} = W_L b^{(L-1)} + b_L = b_L = 0$$

  and we are in the setting of Han et al. (2022). This constrains the cone at the last nonlinear layer to be based at the origin.

- If we wish to consider strong neural collapse (SNC) as defined by Mixon et al. (2022), then [9]

  $$\overline{x^{(L-1)}} = Q^{-1} \sum_{j=1}^{Q} \overline{x_{0,j}^{(L-1)}} = 0,$$

  which implies that $\overline{X^{(L-1)}} \in \mathbb{R}^{Q \times Q}$ cannot be full rank. To conserve injectivity of $\overline{X^{(\tau,L-1)}}$, we must have $b^{(L-1)} \neq 0$, and therefore $\tilde{W}_L \neq W_L$. Indeed, SNC assumes

  $$W_L W_L^T = \sqrt{Q} \left(\mathbf{1}_{Q \times Q} - \frac{1}{Q} u_Q u_Q^T \right), \quad b_L = \frac{1}{Q} u_Q,$$

9. Note that SNC is not reasonable if the penultimate layer features come from a ReLU neural network, as $\overline{x^{(L-1)}} = 0$ would imply that $x_{0,j,i}^{(L-1)} = 0$ for all $j = 1, \cdots, Q$, $i = 1, \cdots, N_j$. Still, the study of the ranks of the matrices and comparison with the first subcase is instructive.

and in that case not only is the last-layer weight matrix not the least squares matrix (35), it is not surjective. To fit this, the constructions in Proposition 3 and Theorem 6 could be modified so that $W^{(L)} \in \mathbb{R}^{Q\times M}$ has rank $Q-1$ while still achieving zero loss:

$$W^{(L)} = \left(\mathbf{1}_{Q\times Q} - B^{(L)}\right) \overline{X^{(\tau,L-1)}}^{+}, \quad b^{(L)} = \frac{1}{Q} u_Q.$$

The remarks in this section indicate that the constructions presented in this paper align well with the framework of neural collapse, at least in certain cases or with minor adaptations. Therefore, employing the truncation map as a tool to investigate the analytic and geometric properties of deep neural networks could also be advantageous in this context.

7. Proof of Proposition 3

To prove the proposition, we use barycentric coordinates. We consider a point $x \in \mathbb{R}^M$ to be represented by

$$x = \sum_{j=1}^{Q} \kappa_j \overline{x_{0,j}} + \tilde{x} \tag{38}$$

where $\tilde{x} \in (\mathrm{span}\{\overline{x_{0,j}}\}_{j=1}^{Q})^{\perp}$, and the means $\overline{x_{0,j}}$ form the vertices of a Q-simplex for which $\overline{x}$ is the barycenter (see Figure 1). Switching to these coordinates can be done when pre-processing the data.

Proof Assume the data $\mathcal{X}_0$ is given in barycentric coordinates. We wish to construct $W^{(\ell)}, b^{(\ell)}$, $\ell = 1, \cdots, Q$, such that $\Delta X_0^{(\tau,Q)} = 0$, and the last layer can be explicitly solved to achieve $X^{(L)} = Y^{ext}$.

Let

$$\theta_* := \theta_0 + \max_j \theta_{*,j},$$

for some $\theta_0 \geq 0$ such that $\theta_* < \pi$. For convenience we also require that θ_0 is such that $\theta_* \geq \frac{\pi}{2}$. Recall from (9)

$$\begin{aligned} P^{(\ell)} &= (W^{(\ell)})^{+} W^{(\ell)} (W^{(\ell-1)})^{+} W^{(\ell-1)} \cdots (W^{(1)})^{+} W^{(1)} \\ &= (W^{(\ell)})^{+} W^{(\ell)} P^{(\ell-1)}. \end{aligned}$$

Induction base $\ell = 1$. If we choose

$$W_1 = W^{(1)} := \mathcal{I}(d_0, d_1) W_{\theta_*} R_1,$$

for W_{θ_*} defined as in (16), and $R_1 \in O(d_0, \mathbb{R})$ such that

$$R_1 f_1 = \frac{u_{d_0}}{|u_{d_0}|},$$

and also

$$b_1 = b^{(1)} := -W_1(\overline{x_{0,1}} + \mu_1 f_1),$$

for some μ_1, then by Lemma 2,

$$\tau_{W^{(1)},b^{(1)}}(x) = \begin{cases} P^{(1)}x, & x \in (\overline{x_{0,1}} + \mu_1 f_1) + \mathfrak{C}_{\theta_1}[f_1], \\ P^{(1)}(\overline{x_{0,1}} + \mu_1 f_1), & x \in (\overline{x_{0,1}} + \mu_1 f_1) + \mathfrak{C}_{\theta_1}[-f_1]. \end{cases}$$

Thus, we wish to find μ_1 such that $\mathcal{X}_{0,1}$ is entirely contained in the backward cone, and $\bigcup_{j\neq 1} \mathcal{X}_{0,j}$ is entirely contained in the forward cone. We have chosen θ_* such that

$$\bigcup_{j\neq 1} B_{4\delta}(\overline{x_{0,j}}) \subset \overline{x_{0,1}} + \mathfrak{C}_{\theta_*}[f_1],$$

and so $\mu_1 < 3\delta$ suffices[10] for

$$\bigcup_{j\neq 1} B_{\delta}(\overline{x_{0,j}}) \subset (\overline{x_{0,1}} + \mu_1 f_1) + \mathfrak{C}_{\theta_*}[f_1].$$

Then, as we have also assumed that $\theta_* \geq \frac{\pi}{2}$, and since in general[11]

$$B_\delta(0) \subset -2\delta f + \mathfrak{C}_{\frac{\pi}{2}}[f]$$

for any $\delta > 0$ and unit vector $f \in \mathbb{R}^{d_0}$, it is sufficient to assume that $\mu_1 > 2\delta$ for

$$B_\delta(\overline{x_{0,1}}) \subset (\overline{x_{0,1}} + \mu_1 f_1) + \mathfrak{C}_{\theta_*}[-f_1].$$

Hence, for $\mu_1 \in (2\delta, 3\delta)$,

$$\tau_{W^{(1)},b^{(1)}}(X_{0,j}) = \begin{cases} P^{(1)}X_{0,j}, & j > 1, \\ P^{(1)}(\overline{x_{0,1}} + \mu_1 f_1)u_{N_1}^T, & j = 1. \end{cases}$$

The induction step $\ell - 1 \to \ell$. Assume

$$X_{0,j}^{(\tau,\ell-1)} = \begin{cases} P^{(\ell-1)}(\overline{x_{0,j}} + \mu_j f_j)u_{N_j}^T, & j \leq \ell - 1, \\ P^{(\ell-1)}X_{0,j}, & j > \ell - 1, \end{cases} \tag{39}$$

for some collection $\{\mu_j\}_{j=1}^{\ell-1} \subset (2\delta, 3\delta)$. We will construct $W^{(\ell)}, b^{(\ell)}$ such that

$$X_{0,j}^{(\tau,\ell)} = \tau_{W^{(\ell)},b^{(\ell)}}(X_{0,j}^{(\tau,\ell-1)}) = \begin{cases} P^{(\ell)}(\overline{x_{0,j}} + \mu_j f_j)u_{N_j}^T, & j < \ell, \\ P^{(\ell)}(\overline{x_{0,\ell}} + \mu_\ell f_\ell)u_{N_\ell}^T, & j = \ell, \\ P^{(\ell)}X_{0,j}, & j > \ell. \end{cases} \tag{40}$$

10. This is not optimal.
11. See appendix.

Let

$$W^{(\ell)} := \mathcal{I}(d_0, d_\ell) W_{\theta_*} R_\ell,$$
$$b^{(\ell)} := -W^{(\ell)} \, (\overline{x_{0,\ell}} + \mu_\ell f_\ell), \tag{41}$$

for $\mathcal{I}(d_0, d_\ell)$ the projection defined in (15), $\mu_\ell \in (2\delta,\, 3\delta)$ and $R_\ell \in O(d_0, \mathbb{R})$ such that

$$R_\ell h_\ell = \frac{u_{d_0}}{|u_{d_0}|}.$$

With these definitions and Lemma 2, we have the following truncation map with its associated cones,

$$\tau_{W^{(\ell)},b^{(\ell)}}(x) = \begin{cases} (W^{(\ell)})^+ W^{(\ell)} \, x, & x \in (\overline{x_{0,\ell}} + \mu_\ell f_\ell) + \mathfrak{C}_{\theta_\ell}[f_\ell], \\ (W^{(\ell)})^+ W^{(\ell)} \, (\overline{x_{0,\ell}} + \mu_\ell f_\ell), & x \in (\overline{x_{0,\ell}} + \mu_\ell f_\ell) + \mathfrak{C}_{\theta_\ell}[-f_\ell]. \end{cases}$$

Just as in the base case, $\mu_\ell < 3\delta$ implies

$$\bigcup_{j \neq \ell} B_\delta(\overline{x_{0,j}}) \subset (\overline{x_{0,\ell}} + \mu_\ell f_\ell) + \mathfrak{C}_{\theta_*}[f_\ell], \tag{42}$$

so that $\bigcup_{j\neq\ell} \mathcal{X}_{0,j}$ sits in the forward cone, and $\mu_\ell > 2\delta$ implies

$$B_\delta(\overline{x_{0,\ell}}) \subset (\overline{x_{0,\ell}} + \mu_\ell f_\ell) + \mathfrak{C}_{\theta_*}[-f_\ell]. \tag{43}$$

so that $\mathcal{X}_{0,\ell}$ sits inside the backward cone. However, now that the data has already been truncated at least once, it takes a little more work to justify how $\tau_{W^{(\ell)},b^{(\ell)}}$ acts on each $X_{0,j}^{(\tau,\ell-1)}$.

For $j < \ell$, we have from (39) that

$$\mathcal{X}_{0,j}^{(\tau,\ell-1)} = P^{(\ell-1)}(\overline{x_{0,j}} + \mu_j f_j),$$

and since $W^{(\ell)} P^{(\ell-1)} = W^{(\ell)}$ by (11) in Lemma 1, we can use (17) to get

$$\tau_{W^{(\ell)},b^{(\ell)}}(P^{(\ell-1)}(\overline{x_{0,j}} + \mu_j f_j)) = \tau_{W^{(\ell)},b^{(\ell)}}(\overline{x_{0,j}} + \mu_j f_j).$$

We claim that

$$(\overline{x_{0,j}} + \mu_j f_j) \in (\overline{x_{0,\ell}} + \mu_\ell f_\ell) + \mathfrak{C}_{\theta_*}[f_\ell]. \tag{44}$$

Indeed, note that for all $j' = 1, \cdots, Q$,

$$\overline{x} = \overline{x_{0,j'}} + \left|\overline{x_{0,j'}} - \overline{x}\right| f_{j'}, \tag{45}$$

so it is clear that

$$\overline{x} \in \overline{x_{0,\ell}} + \mathfrak{C}_{\theta_*}[f_\ell],$$

and if $\mu < |\overline{x_{0,\ell}} - \overline{x}|$, then also

$$\overline{x} \in (\overline{x_{0,\ell}} + \mu f_\ell) + \mathfrak{C}_{\theta_*}[f_\ell]. \tag{46}$$

By assumption (18),

$$\delta < \frac{1}{4} \min_j |\overline{x_{0,j}} - \overline{x}| \,,$$

so in particular

$$\mu_\ell < 3\delta < |\overline{x_{0,\ell}} - \overline{x}| \,.$$

Again by (18) and (45),

$$\mu_j < 3\delta < |\overline{x_{0,j}} - \overline{x}|$$

and $(\overline{x_{0,j}} + \mu_j f_j)$ is in the line segment between $\overline{x_{0,j}}$ and $\overline{x}$, which is entirely contained in the forward cone by (42), (46) and the convexity of the cone. Having proved (44),

$$\begin{aligned} X_{0,j}^{(\tau,\ell)} &= \tau_{W^{(\ell)},b^{(\ell)}} \left(X_{0,j}^{(\tau,\ell-1)} \right) \\ &= (W^{(\ell)})^+ W^{(\ell)} X_{0,j}^{(\tau,\ell-1)} \\ &= (W^{(\ell)})^+ W^{(\ell)} P^{(\ell-1)} (\overline{x_{0,j}} + \mu_j f_j) u_{N_j}^T \\ &= P^{(\ell)} (\overline{x_{0,j}} + \mu_j f_j) u_{N_j}^T . \end{aligned}$$

For $j = \ell$, we have assumed that

$$\mathcal{X}_{0,\ell}^{(\tau,\ell-1)} = P^{(\ell-1)} \mathcal{X}_{0,\ell},$$

and once again by (11),

$$\tau_{W^{(\ell)},b^{(\ell)}} \left(X_{0,\ell}^{(\tau,\ell-1)} \right) = \tau_{W^{(\ell)},b^{(\ell)}} (X_{0,\ell})$$

and we can use (43) in

$$\begin{aligned} X_{0,\ell}^{(\tau,\ell)} &= \tau_{W^{(\ell)},b^{(\ell)}} \left(X_{0,\ell}^{(\tau,\ell-1)} \right) \\ &= (W^{(\ell)})^+ W^{(\ell)} (\overline{x_{0,\ell}} + \mu_\ell f_\ell) u_{N_\ell}^T \\ &\overset{(*)}{=} P^{(\ell)} (\overline{x_{0,\ell}} + \mu_\ell f_\ell) u_{N_\ell}^T , \end{aligned}$$

where $(*)$ follows from Lemma 1.

Similarly, for $j > \ell$,

$$\mathcal{X}_{0,j}^{(\tau,\ell-1)} = P^{(\ell-1)} \mathcal{X}_{0,j},$$

and we can use (11) and (42) to compute

$$\begin{aligned} X_{0,j}^{(\tau,\ell)} &= \tau_{W^{(\ell)},b^{(\ell)}}\left(X_{0,j}^{(\tau,\ell-1)}\right) \\ &= (W^{(\ell)})^{+}W^{(\ell)}X_{0,j}^{(\tau,\ell-1)} \\ &= (W^{(\ell)})^{+}W^{(\ell)}P^{(\ell-1)}X_{0,j} \\ &= P^{(\ell)}X_{0,j}, \end{aligned}$$

and we have proved (40).

Conclusion of proof. After $L-1=Q$ steps,

$$X_{0,j}^{(\tau,Q)} = \overline{X_{0,j}^{(\tau,Q)}} = P^{(Q)}(\overline{x_{0,j}} - \mu_j f_j)u_{N_j}^T,$$

so $\Delta X_{0,j}^{(\tau,Q)} = 0$. In the last layer, we have from (8)

$$X^{(L)} = W^{(Q+1)}\overline{X_0^{(\tau,Q)}} + B^{(Q+1)}.$$

It is easy to check that if $(\overline{X_0^{red}})^{(\tau,Q)}$ is full rank, then

$$\begin{aligned} W^{(Q+1)} &= Y((\overline{X_0^{red}})^{(\tau,Q)})^{+}, \\ b^{(Q+1)} &= 0, \end{aligned}$$

solve

$$W^{(Q+1)}\overline{X_0^{(\tau,Q)}} + B^{(Q+1)} = Y^{ext}, \tag{47}$$

so that the cumulative parameters (41) and (47) define a neural network with weights and biases $(W_i^{**}, b_i^{**})_{i=1}^{Q+1}$ such that

$$X^{(L)} = Y^{ext}$$

and

$$\mathcal{C}[(W_i^{**}, b_i^{**})_{i=1}^{Q+1}] = 0.$$

It remains to check that $\overline{X_0^{red}}^{(\tau,Q)} \in \mathbb{R}^{M\times Q}$ is indeed full rank, which is the same as checking that the set $\left\{P^{(Q)}(\overline{x_{0,j}} - \mu_j f_j)\right\}_{j=1}^{Q}$ is linearly independent. First, note that the set of vectors $\{(\overline{x_{0,j}} - \mu_j f_j)\}_{j=1}^{Q}$ is linearly independent: We have assumed that $\overline{X_0^{red}}$ is full rank, so the means $\overline{x_{0,j}}$ form the vertices of a Q-simplex which spans a Q-dimensional subspace of $\mathbb{R}^M$. Each $-\mu_j f_j$ simply shifts a vertex of this simplex towards the barycenter $\overline{x}$ in such a way that $\{(\overline{x_{0,j}} - \mu_j f_j)\}_{j=1}^{Q}$ forms a smaller simplex spanning the same Q-dimensional subspace.

Next, we need $P^{(Q)}$ to be suitably rank preserving. This can be done by changing the matrices $\tilde{R}$ and R_Q in the definitions of W_{θ_*} and $W^{(Q)}$, respectively, as we explain now. From (10) in Lemma 1,

$$\begin{aligned} P^{(Q)} &= (W^{(Q)})^{+} W^{(Q)} \\ &= (W^{(Q)})^T \left(W^{(Q)} (W^{(Q)})^T \right)^{-1} W^{(Q)} \\ &= (W_{\theta_*} R_Q)^T \mathcal{I}(d_0, d_Q)^T (W^{(Q)} W^{(Q)\,T})^{-1} \mathcal{I}(d_0, d_Q) W_{\theta_*} R_Q, \end{aligned}$$

then $W_{\theta_*} R_Q \in GL(d_0, \mathbb{R})$, and $\mathcal{I}(d_0, d_Q)^T$ is an inclusion, and so

$$(W_{\theta_*} R_Q)^T \mathcal{I}(d_0, d_Q)^T (W^{(Q)} W^{(Q)\,T})^{-1}$$

does not change the rank of any matrix on which it acts.

We now turn to the action of $\mathcal{I}(d_0, d_Q) W_{\theta_*} R_Q$. As we are working in barycentric coordinates (38), we only care about the action of $P^{(Q)}$ on the first Q coordinates of any vector, and so restrict our attention to the first Q columns of $\mathcal{I}(d_0, d_Q) W_{\theta_*} R_Q$. Moreover, $\mathcal{I}(d_0, d_Q)$ acts on $W_{\theta_*} R_Q$ by cutting off all but the first $d_Q \geq Q$ rows. Altogether, it is sufficient that the first $Q \times Q$ block of $W_{\theta_*} R_Q$ be invertible. Using the Laplace cofactor expansion of the determinant, it can be shown that there exists a permutation of the rows of $W_{\theta_*} R_Q$ that achieves this.

Let P be the elementary matrix that realizes this permutation, and note that $P \in O(d_0, \mathbb{R})$. Then letting $\lambda := \lambda(\theta_*, \theta_{d_0})$ in the following for notational convenience, and expanding the definition of W_{θ_*} from (16),

$$\begin{aligned} P W_{\theta_*} R_Q &= P \tilde{R} \operatorname{diag}(1, \lambda, \cdots, \lambda) \tilde{R}^T R_Q \\ &= P \tilde{R} \operatorname{diag}(1, \lambda, \cdots, \lambda) \tilde{R}^T (P^T P) R_Q \\ &= (P \tilde{R}) \operatorname{diag}(1, \lambda, \cdots, \lambda) (P \tilde{R})^T (P R_Q) \end{aligned}$$

and so the permutation needed amounts to permuting the rows of $\tilde{R}$ and R_Q. Recall the definitions

$$\tilde{R} e_1^{d_0} = \frac{u_{d_0}}{|u_{d_0}|} = R_Q f_Q. \tag{48}$$

Clearly, $P u_{d_0} = u_{d_0}$, and so we can pick $\tilde{R}$ and R_Q satisfying (48) and such that the block $(W_{\theta_*} R_Q)_{Q \times Q}$ is invertible, and $P^{(Q)}$ restricted to span $\{\overline{x_{0,j}}\}_{j=1}^{Q}$ is injective.

Finally, note that the global minimum we have constructed is degenerate, since we have a family of cumulative bias terms $(b_*^{(\ell)}[\mu_\ell])_{\ell=1}^{Q}$ parametrized by $(\mu_\ell)_{\ell=1}^{Q} \subset (2\delta, 3\delta)^Q$. ■

Acknowledgments: T.C. gratefully acknowledges support by the NSF through the grant DMS-2009800, and the RTG Grant DMS-1840314 - *Analysis of PDE*. P.M.E. was supported by NSF grant DMS-2009800 through T.C., and UT Austin's GS Summer fellowship.

Appendix A. Geometry

As usual, we define the angle between two vectors $u, v, \in \mathbb{R}^n$ to be

$$\angle(u, v) := \arccos \frac{\langle u, v\rangle}{\|u\| \, \|v\|}.$$

Then

$$\angle(u, v) \leq \theta$$

is equivalent to

$$\frac{\langle u, v\rangle}{\|u\| \, \|v\|} \geq \cos\theta.$$

A cross section of the largest cone centered around the diagonal in $\mathbb{R}^n$ is a closed $(n-1)$-dimensional disk inscribed into the n-simplex. The angle θ_n of such a cone can be obtained by analyzing a point in its boundary, such as the barycenter of any one of the faces. That is, if we choose the j-th face, we consider

$$v_j = \sum_{i \neq j} \frac{1}{n-1} e_i$$

In this case

$$\|u_n\| = \sqrt{n}, \quad \|v_j\| = \frac{1}{\sqrt{n-1}},$$
$$\langle u_n, v_j\rangle = (n-1)\frac{1}{n-1} = 1,$$

so that

$$\angle(u_n, v_j) = \arccos \frac{\sqrt{n-1}}{\sqrt{n}},$$

giving the expression (13) for θ_n.

In the proof of Proposition 3, we make the assumption $\theta_* \geq \frac{\pi}{2}$. Note that a cone based at the origin with axis some unit vector $f \in \mathbb{R}^n$ will contain the ball $B_\delta(2\delta f)$ if it contains

$$\delta f + \delta f^\perp$$

for any $f^\perp$ a unit vector perpendicular to f. The smallest angle for which this holds is given by

$$\begin{aligned}
\cos(\theta) &= \frac{\langle \delta f, \delta f + \delta f^\perp\rangle}{\|\delta f\| \, \|\delta f + \delta f^\perp\|} \\
&= \frac{\|\delta f\|}{\sqrt{\|\delta f\|^2 + \|\delta f^\perp\|^2}} \\
&= \frac{1}{\sqrt{2}},
\end{aligned}$$

or $\theta = \frac{\pi}{2}$.

References

E. M. Achour, F. Malgouyres, and S. Gerchinovitz. The loss landscape of deep linear neural networks: a second-order analysis. *Journal of Machine Learning Research*, 25(242):1–76, 2024.

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

S. L. Campbell and C. D. Meyer. *Generalized inverses of linear transformations*. SIAM, 2009.

T. Chen and P. M. Ewald. Geometric structure of deep learning networks and construction of global $\mathcal{L}^2$ minimizers, 2023a. arXiv:2309.10639

T. Chen and P. M. Ewald. Geometric structure of shallow neural networks and constructive $\mathcal{L}^2$ cost minimization, 2023b. arXiv:2309.10370

T. Chen and P. M. Ewald. On non-approximability of zero loss global $\mathcal{L}^2$ minimizers by gradient descent in deep learning. *Theor. Appl. Mech.*, 52(1), 67–73, 2025. arXiv:2311.07065.

T. Chen and A. G. Moore. Zero loss guarantees and explicit minimizers for generic overparametrized deep learning networks, 2025. arXiv:2502.14114.

T. Chen, C.-K. K. Chien, P. M. Ewald, and A. G. Moore. Architecture independent generalization bounds for overparametrized deep relu networks, 2025. arXiv:2504.05695.

A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204. PMLR, 2015.

P. M. Ewald. Explicit neural network classifiers for non-separable data, 2025. arXiv:2504.18710

J. E. Grigsby, K. Lindsey, and D. Rolnick. Hidden symmetries of relu networks. In *International Conference on Machine Learning*, pages 11734–11760. PMLR, 2023.

J. E. Grigsby, K. Lindsey, R. Meyerhoff, and C. Wu. Functional dimension of feedforward relu neural networks, 2022. arXiv:2209.04036.

X. Han, V. Papyan, and D. L. Donoho. Neural collapse under MSE loss: Proximity to and dynamics on the central path. In *International Conference on Learning Representations*, 2022.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

K. Kawaguchi. Deep learning without poor local minima. *Advances in Neural Information Processing Systems*, 29, 2016.

J. Lee, J. Sohl-Dickstein, J. Pennington, R. Novak, S. Schoenholz, and Y. Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

D. G. Mixon, H. Parshall, and J. Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 2022.

V. Papyan, X. Han, and D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117 (40):24652–24663, 2020.

M. Phuong and C. H. Lampert. The inductive bias of relu networks on orthogonally separable data. In *International Conference on Learning Representations*, 2020.

Y. Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *International Conference on Machine Learning*, pages 3404–3413. PMLR, 2017.

V. Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

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