A. Isomorphism under scaling

Lemma 1. *Given a fully connected ReLU network $\mathcal{N}$, the network $s_{z,c}(\mathcal{N})$ is isomorphic to $\mathcal{N}$ for every neuron z and constant $c > 0$.*

Suppose that $z = z_i^k$ is the ith neuron in layer k. Then, for each neuron z_j^{k+1} in layer $k+1$ of the network $\mathcal{N}$, we have:

$$\begin{aligned} z_j^{k+1}(\mathbf{x}) &= \sum_{i=1}^{n_k} \mathbf{W}_{ij}^k \operatorname{ReLU}(z_i^k(\mathbf{x})) + \mathbf{b}_j^{k+1} \\ &= \sum_{i=1}^{n_k} \mathbf{W}_{ij}^k \operatorname{ReLU}\left(\sum_{h=1}^{n_{k-1}} \mathbf{W}_{hi}^{k-1} \operatorname{ReLU}(\right. \\ &\qquad \left. z_h^{k-1}(\mathbf{x})) + \mathbf{b}_i^k \right) + \mathbf{b}_j^{k+1}. \end{aligned} \tag{1}$$

By comparison, in network $s_{z,c}(\mathcal{N})$, we have:

$$\begin{aligned} z_j^{k+1}(\mathbf{x}) &= \sum_{i=1}^{n_k} \frac{1}{c} \mathbf{W}_{ij}^k \operatorname{ReLU}\left(\sum_{h=1}^{n_{k-1}} c\mathbf{W}_{hi}^{k-1} \operatorname{ReLU}(\right. \\ &\qquad \left. z_h^{k-1}(\mathbf{x})) + c\mathbf{b}_i^k \right) + \mathbf{b}_j^{k+1} \\ &= \sum_{i=1}^{n_k} \mathbf{W}_{ij}^k \operatorname{ReLU}\left(\sum_{h=1}^{n_{k-1}} \mathbf{W}_{hi}^{k-1} \operatorname{ReLU}(\right. \\ &\qquad \left. z_h^{k-1}(\mathbf{x})) + \mathbf{b}_i^k \right) + \mathbf{b}_j^{k+1}. \end{aligned} \tag{2}$$

where we used the property that $\operatorname{ReLU}(cx) = c \operatorname{ReLU}(x)$ for any $c > 0$.

As expressions (1) and (2) are equal, we conclude that $s_{z,c}(\mathcal{N})$ is isomorphic to $\mathcal{N}$.

B. Proof of Theorem 1

It is observed in Hanin & Rolnick (2019b) that B_z cannot bend except at points of intersection with $B_{z'}$ for z' in an earlier layer than z. We now prove the converse. Suppose that neurons z, z' are such that z' lies in an earlier layer than z. Consider a point $\mathbf{p}$ of intersection between B_z and $B_{z'}$, and suppose that $\mathbf{p}_1$ and $\mathbf{p}_2$ are in an arbitrarily small neighborhood of $\mathbf{p}$, lying on opposite sides of $B_{z'}$. It suffices to prove that $\nabla z(\mathbf{p}_1) \neq \nabla z(\mathbf{p}_2)$, and therefore that B_z bends as it intersects $B_{z'}$.

By the Linear Regions Assumption, $\mathcal{N}(\mathbf{x})$ computes different functions on the two sides of $B_{z'}$. Since $\mathcal{N}(\mathbf{x})$ is continuous, $\nabla\mathcal{N}(\mathbf{x})$ must differ on the two sides of $B_{z'}$; that is, $\nabla\mathcal{N}(\mathbf{p}_1) \neq \nabla\mathcal{N}(\mathbf{p}_2)$. Suppose that $z = z_j^k$ lies in layer k; then there exists some neuron z_ℓ^k in layer k such that $\nabla z_\ell^k(\mathbf{p}_1) \neq \nabla z_\ell^k(\mathbf{p}_2)$. If $j = \ell$, we are done. Otherwise, observe that

$$z_\ell^k(\mathbf{x}) = \sum_{h=1}^{n_{k-1}} \mathbf{W}_{h\ell}^k \operatorname{ReLU}(z_h^{k-1}(\mathbf{x})) + \mathbf{b}_\ell^k.$$

Consider the $n_{\text{in}} \times n_{k-1}$ matrix $M(\mathbf{x})$ with columns $\nabla \operatorname{ReLU}(z_h^{k-1}(\mathbf{x}))$ indexed by h. As $\nabla z_\ell^k(\mathbf{x})$ is a linear combination of these columns, we conclude that $M(\mathbf{p}_1) \neq M(\mathbf{p}_2)$. Note that $\nabla z(\mathbf{x}) = \nabla z_j^k(\mathbf{x})$ is a linear combination of the columns of $M(\mathbf{x})$ with coefficients $\mathbf{W}_{hj}^k$. Since $M(\mathbf{p}_1) \neq M(\mathbf{p}_2)$, we conclude that with probability 1 over the choice of $\mathbf{W}_{hj}^k$, we must have $\nabla z(\mathbf{p}_1) \neq \nabla z(\mathbf{p}_2)$, as desired.

C. Proof of Theorem 4

In this proof, we will show how the information we are given by the assumptions of the theorem is enough to recover the weights and biases for each neuron z in layer k. We will proceed for each z individually, progressively learning weights between z and each of the neurons in the preceding layer (though for skip connections this procedure could also easily be generalized to learn weights from z to earlier layers).

For each of the points $\mathbf{p}_i \in A_z$, suppose that H_i is the local hyperplane associated with $\mathbf{p}_i$ on boundary B_z. The gradient $\nabla z(\mathbf{p}_i)$ at $\mathbf{p}_i$ is orthogonal to H_i, and we thus already know the direction of the gradient, but its magnitude is unknown to us. We will proceed in order through the points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m$, with the goal of identifying $\nabla z(\mathbf{p}_i)$ for each $\mathbf{p}_i$, up to a single scaling factor, as this computation will end up giving us the incoming weights for z.

We begin with $\mathbf{p}_1$ by assigning $\nabla z(\mathbf{p}_1)$ arbitrarily to either one of the two unit vectors orthogonal to H_i. Due to scaling invariance (Lemma 1), the weights of $\mathcal{N}$ can be rescaled without changing the function so that $\nabla z(\mathbf{p}_i)$ is multiplied by any positive constant. Therefore, our arbitrary choice can be wrong at most in its sign, and we need not determine the sign at this stage. Now, suppose towards induction that we have identified $\nabla z(\mathbf{p}_i)$ (up to sign) for $i = 1, \ldots, s-1$. We wish to identify $\nabla z(\mathbf{p}_s)$.

By assumption (ii), there exists a precursor $\mathbf{p}_r$ to $\mathbf{p}_s$ such that H_r and H_s intersect on a boundary $B_{z'}$. Let $\mathbf{v}_r = t_z \nabla z(\mathbf{p}_r)$ be our estimate of $\nabla z(\mathbf{p}_r)$, for unknown sign $t_z \in \{+1, -1\}$. Let $\mathbf{v}_s$ be a unit normal vector to H_s, so that $\mathbf{v}_s = c t_z \nabla z(\mathbf{p}_s)$ for some unknown constant c. We pick the sign of $\mathbf{v}_s$ so that it has the same orientation as $\mathbf{v}_r$ with respect to the surface B_z, and thus $c > 0$. Finally, let $\mathbf{v} = t_{z'} \nabla z'(\mathbf{p}_r) = t_{z'} \nabla z'(\mathbf{p}_s)$ be our estimate of the gradient of z'; where $t_{z'} \in \{+1, -1\}$ is also an unknown sign (recall that since z' is in layer $k-1$ we know its gradient up to sign). We will use $\mathbf{v}$ and $\mathbf{v}_r$ to identify $\mathbf{v}_s$.

Suppose that $z = z_j^k$ is the jth neuron in layer k and that $z' = z_h^{k-1}$ is the hth neuron in layer $k-1$. Recall that

$$z(\mathbf{x}) = z_j^k(\mathbf{x}) = \sum_{i=1}^{n_{k-1}} \mathbf{W}_{ij}^k \, \mathrm{ReLU}(z_i^{k-1}(\mathbf{x})) + \mathbf{b}_j^k. \quad (3)$$

As $B_{z'}$ is the boundary between inputs for which $z' = z_h^{k-1}$ is active and inactive, $\mathrm{ReLU}(z_h^{k-1}(\mathbf{x}))$ must equal zero either (Case 1) on H_r or (Case 2) on H_s.

In Case 1, we have

$$\nabla z(\mathbf{p}_s) - \nabla z(\mathbf{p}_r) = \mathbf{W}_{hj}^k \nabla z'(\mathbf{p}_r),$$

or equivalently:

$$c t_z \mathbf{v}_s - t_z \mathbf{v}_r = \mathbf{W}_{hj}^k t_{z'} \mathbf{v},$$

which gives us the equation:

$$c\mathbf{v}_s - \mathbf{v}_r = \mathbf{W}_{hj}^k t_z t_{z'} \mathbf{v}.$$

Since we know the vectors $\mathbf{v}_s, \mathbf{v}_r, \mathbf{v}$, we are able to deduce the constant c.

A similar equation arises in Case 2:

$$\mathbf{v}_r - c\mathbf{v}_s = \mathbf{W}_{hj}^k t_z t_{z'} \mathbf{v},$$

giving rise to the same value of c. We thus may complete our induction. In the process, observe that we have calculated a constant $\mathbf{W}_{hj}^k t_z t_{z'} t'$, where the sign t' is $+1$ in Case 1 and -1 in Case 2. Note that $t_{z'} t'$ can be calculated based on whether $\mathbf{v}$ points towards $\mathbf{p}_r$ or $\mathbf{p}_s$. Therefore, we have obtained $\mathbf{W}_{hj}^k t_z$, which is exactly the weight (up to z-dependent sign) that we wished to find. Once we have all weights incoming to z (up to sign), it is simple to identify the bias for this neuron (up to sign) by calculating the equation of any known local hyperplane for B_z and using the known weights and biases from earlier layers.

To complete the proof, we must now also calculate the correct signs $t_{z'}$ of the neurons in layer $k-1$. Pick some $z = z_j^k$ in layer k and observe that for all points $\mathbf{p}_s \in A_z$ there corresponds an equation, obtained by taking gradients in equation (3):

$$\nabla z_j^k(\mathbf{p}_s) = \sum_{i=1}^{n_{k-1}} \mathbf{W}_{ij}^k \mathbb{1}_{i,s} \nabla z_i^{k-1}(\mathbf{p}_s),$$

where $\mathbb{1}_{i,s}$ equals 1 if $\mathbf{p}_s$ is on the active side of $B_{z_i^{k-1}}$. We can substitute in our (sign-unknown) values for these various quantities:

$$t_z \mathbf{v}_s = \sum_{i=1}^{n_{k-1}} \mathbf{W}_{ij}^k \mathbb{1}_{i,s} t_{z_i^{k-1}} \mathbf{v}_i.$$

Now, we may estimate $\mathbb{1}_{i,s}$ by a function $\mathbb{1}'_{i,s}$ that is 1 if $\mathbf{p}_s$ and $\mathbf{v}_i$ are on the same side of $B_{z_i^{k-1}}$. This estimate will be wrong exactly when $t_{z_i^{k-1}} = -1$. Thus, $\mathbb{1}_{i,s} = (1 + t_{z_i^{k-1}} \mathbb{1}'_{i,s})/2$, giving us the equation:

$$\begin{aligned} t_z \mathbf{v}_s &= \sum_{i=1}^{n_{k-1}} \mathbf{W}_{ij}^k \frac{1 + t_{z_i^{k-1}} \mathbb{1}'_{i,s}}{2} t_{z_i^{k-1}} \mathbf{v}_i \\ &= \frac{1}{2} \sum_{i=1}^{n_{k-1}} \mathbf{W}_{ij}^k (t_{z_i^{k-1}} + \mathbb{1}'_{i,s}) \mathbf{v}_i \end{aligned}$$

All the terms of this equation are known, with the exception of t_z and the n_{k-1} variables $t_{z_i^{k-1}}$ – giving us a linear system in $n_{k-1} + 1$ variables. For a given z_j^k, there are n_{k-1} different $\mathbf{p}_s$ representing the intersections with $B_{z'}$ for each z' in layer $k-1$; choosing these $\mathbf{p}_s$ should in general give linearly independent constraints. Moreover, the equation is in fact a vector equality with dimension n_{in}; hence, it is a highly overconstrained system, enabling us to identify the signs $t_{z_i^{k-1}}$ for each z_i^{k-1}. This completes the proof of the theorem.