

---

# Functional Equivalence and Path Connectivity of Reducible Hyperbolic Tangent Networks

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

Understanding the learning process of artificial neural networks requires clarifying the structure of the parameter space within which learning takes place. A neural network parameter’s *functional equivalence class* is the set of parameters implementing the same input–output function. For many architectures, almost all parameters have a simple and well-documented functional equivalence class. However, there is also a vanishing minority of *reducible* parameters, with richer functional equivalence classes caused by redundancies among the network’s units.

In this paper, we give an algorithmic characterisation of unit redundancies and reducible functional equivalence classes for a single-hidden-layer hyperbolic tangent architecture. We show that such functional equivalence classes are piecewise-linear path-connected sets, and that for parameters with a majority of redundant units, the sets have a diameter of at most 7 linear segments.

## 1 Introduction

Deep learning algorithms construct a parameter for an artificial neural network architecture through a local search in the high-dimensional parameter space. This search is guided by the topography of some loss landscape. This topography is in turn determined by the relationship between neural network parameters and neural network input–output functions. Thus, understanding the relationship between these parameters and functions is key to understanding deep learning.

It is well known that neural network parameters often fail to uniquely determine an input–output function. For example, exchanging weights between two adjacent hidden units generally preserves functional equivalence (Hecht-Nielsen, 1990). For many architectures, almost all parameters have a simple class of functionally equivalent parameters. These classes have been characterised for multi-layer feed-forward architectures with various nonlinearities (e.g., Sussmann, 1992; Albertini et al., 1993; Kůrková and Kainen, 1994; Phuong and Lampert, 2020; Vlačić and Bölcskei, 2021).

However, all existing work on functional equivalence excludes from consideration certain measure zero sets of parameters, for which the functional equivalence classes may be richer. One such family of parameters is the so-called *reducible parameters*. These parameters display certain structural redundancies, such that the same function could be implemented with fewer hidden units (Sussmann, 1992; Vlačić and Bölcskei, 2021), leading to a richer functional equivalence class.

Reducible parameters may play an important role in deep learning. While randomly selected parameters are reducible with probability zero, learned parameters are not random. Structural redundancy has appealing parsimony connotations, and is also related to information singularities in statistical learning (Fukumizu, 1996). Moreover, the structure of functional equivalence classes has implications for the topography of the loss landscape, and, therefore, for the dynamics of learning.

In this paper, we study functional equivalence classes for single-hidden-layer networks with the hyperbolic tangent nonlinearity, building on the foundational work of [Sussmann \(1992\)](#) on reducibility in this setting. While this architecture is not immediately relevant to modern deep learning, structural redundancy has unresearched implications for functional equivalence in *all* architectures. A comprehensive investigation of this simple case is a first step in this research direction. To this end, we offer the following theoretical contributions.

1. In Section 4, we give a formal algorithm producing a canonical representative parameter from any functional equivalence class, by systematically eliminating all sources of structural redundancy. This extends prior algorithms that only handle irreducible parameters.
2. In Section 5, we invert this canonicalisation algorithm to characterise the functional equivalence class of any parameter as a union of simple parameter manifolds. This characterisation extends the well-known result for irreducible parameters.
3. We show that in the reducible case, the functional equivalence class is a piecewise-linear path-connected set—that is, any two functionally equivalent reducible parameters are connected by a piecewise linear path comprising only equivalent parameters (Theorem 6.1).
4. We show that if a parameter has a high degree of reducibility (in particular, if the same function can be implemented using half of the available hidden units), then the number of linear segments required to connect any two equivalent parameters is at most 7 (Theorem 6.3).

In Section 7, we discuss the implications of these results for an understanding of the structure of the parameter space, and outline directions for future work including extensions to modern architectures.

## 2 Related Work

[Sussmann \(1992\)](#) studied functional equivalence in single-hidden-layer hyperbolic tangent networks, showing that two irreducible parameters are functionally equivalent if and only if they are related by simple operations of exchanging and negating the weights of hidden units. This result was later extended to architectures with a broader class of nonlinearities ([Albertini et al., 1993](#); [Kůrková and Kainen, 1994](#)), to architectures with multiple hidden layers ([Fefferman and Markel, 1993](#); [Fefferman, 1994](#)), and to certain recurrent architectures ([Albertini and Sontag, 1992, 1993a,b,c](#)). More recently, similar results have been found for ReLU networks ([Phuong and Lampert, 2020](#); [Bona-Pellissier et al., 2021](#); [Stock and Gribonval, 2022](#)), and [Vlačić and Bölskei \(2021, 2022\)](#) have generalised Sussmann’s results to a very general class of architectures and nonlinearities. However, all of these results have come at the expense of excluding from consideration certain measure zero subsets of parameters with richer functional equivalence classes.

A similar line of work has documented the global symmetries of the parameter space—bulk transformations of the entire parameter space that preserve all implemented functions. The search for such symmetries was launched by [Hecht-Nielsen \(1990\)](#). [Chen et al. \(1993\)](#), also [Chen and Hecht-Nielsen, 1991](#)) showed that in the case of multi-layer hyperbolic tangent networks, all analytic symmetries are generated by unit exchanges and negations. [Rüger and Ossen \(1997\)](#) extended this result to additional sigmoidal nonlinearities. The analyticity condition excludes discontinuous symmetries acting selectively on, say, reducible parameters with richer equivalence classes ([Chen et al., 1993](#)).

[Rüger and Ossen \(1997\)](#) provide a canonicalisation algorithm. Their algorithm negates each hidden unit’s weights until the bias is positive, and then sorts each hidden layer’s units into non-descending order by bias weight. This algorithm is invariant precisely to the exchanges and negations mentioned above, but fails to properly canonicalise equivalent parameters that differ in more complex ways.

To our knowledge there is only one line of work bearing directly on the topic of the functional equivalence classes of reducible parameters. [Fukumizu and Amari \(2000\)](#) and [Fukumizu et al. \(2019\)](#) have catalogued methods of adding a single hidden unit to a neural network while preserving the network’s function, and [Şimşek et al. \(2021\)](#) have extended this work to consider the addition of multiple hidden units. Though derived under a distinct framing, it turns out that the subsets of parameter space accessible by such unit additions correspond to functional equivalence classes, similar to those we study (though in a slightly different architecture). We note these similarities, especially regarding our contributions (2) and (3), in Remarks 5.4 and 5.5 and Remark 6.2.

### 86 3 Preliminaries

87 We consider a family of fully-connected, feed-forward neural network architectures with a single  
 88 input unit, a single biased output unit, and a single hidden layer of  $h \in \mathbb{N}$  biased hidden units with  
 89 the hyperbolic tangent nonlinearity  $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$ . Such an architecture has a  
 90 parameter space  $\mathcal{W}_h = \mathbb{R}^{3h+1}$ . Our results generalise directly to networks with multi-dimensional  
 91 inputs and outputs, as detailed in Appendix A.

92 The weights and biases of the network's units are encoded in the parameter vector in the format  
 93  $(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h$  where for each hidden unit  $i = 1, \dots, h$  there is an *outgoing*  
 94 *weight*  $a_i \in \mathbb{R}$ , an *incoming weight*  $b_i \in \mathbb{R}$ , and a *bias*  $c_i \in \mathbb{R}$ , and  $d \in \mathbb{R}$  is an *output unit bias*.  
 95 Thus each parameter  $w = (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h$  indexes a mathematical function  
 96  $f_w : \mathbb{R} \rightarrow \mathbb{R}$  defined as follows:

$$f_w(x) = d + \sum_{i=1}^h a_i \tanh(b_i x + c_i).$$

97 Two parameters  $w \in \mathcal{W}_h, w' \in \mathcal{W}_{h'}$  are *functionally equivalent* if and only if  $f_w = f_{w'}$  as functions  
 98 on  $\mathbb{R}$  (that is,  $\forall x \in \mathbb{R}, f_w(x) = f_{w'}(x)$ ). Functional equivalence is of course an equivalence relation  
 99 on  $\mathcal{W}_h$ . Given a parameter  $w \in \mathcal{W}_h$ , the *functional equivalence class* of  $w$ , denoted  $\mathfrak{F}[w]$ , is the set  
 100 of all parameters in  $\mathcal{W}_h$  that are functionally equivalent to  $w$ :

$$\mathfrak{F}[w] = \{w' \in \mathcal{W}_h \mid f_w = f_{w'}\}.$$

101 For this family of architectures, the functional equivalence class of almost all parameters is a discrete  
 102 set fully characterised by simple *unit negation and exchange transformations*  $\sigma_i, \tau_{i,j} : \mathcal{W}_h \rightarrow \mathcal{W}_h$   
 103 for  $i, j = 1, \dots, h$ , where

$$\begin{aligned} \sigma_i(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) &= (a_1, b_1, c_1, \dots, -a_i, -b_i, -c_i, \dots, a_h, b_h, c_h, d) \\ \tau_{i,j}(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) &= (a_1, b_1, c_1, \dots, c_{i-1}, a_j, b_j, c_j, a_{i+1}, \\ &\quad \dots, c_{j-1}, a_i, b_i, c_i, a_{j+1}, \dots, a_h, b_h, c_h, d). \end{aligned}$$

104 More formally, these transformations generate the full functional equivalence class for all so-called  
 105 irreducible parameters (Sussmann, 1992). A parameter  $w = (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h$  is  
 106 *reducible* if and only if it satisfies any of the following conditions (otherwise,  $w$  is *irreducible*):

- 107 (i)  $a_i = 0$  for some  $i$ , or
- 108 (ii)  $b_i = 0$  for some  $i$ , or
- 109 (iii)  $(b_i, c_i) = (b_j, c_j)$  for some  $i \neq j$ , or
- 110 (iv)  $(b_i, c_i) = (-b_j, -c_j)$  for some  $i \neq j$ .

111 Sussmann (1992) also showed that in this family of architectures, reducibility corresponds to *non-*  
 112 *minimality*: a parameter  $w \in \mathcal{W}_h$  is reducible if and only if  $w$  is functionally equivalent to some  
 113  $w' \in \mathcal{W}_{h'}$  with fewer hidden units  $h' < h$ . We define the *rank* of  $w$ , denoted  $\text{rank}(w)$ , as the  
 114 minimal number of hidden units required to implement  $f_w$ :

$$\text{rank}(w) = \min \{h' \in \mathbb{N} \mid \exists w' \in \mathcal{W}_{h'}; f_w = f_{w'}\}.$$

115 Finally, we make use of the following notions of connectivity for a set of parameters. Given a set  
 116  $W \subseteq \mathcal{W}_h$ , define a *piecewise linear path in  $W$*  as a continuous function  $\rho : [0, 1] \rightarrow W$  comprising a  
 117 finite number of linear segments. Two parameters  $w, w' \in \mathcal{W}_h$  are *piecewise-linear path-connected*  
 118 *in  $W$* , denoted  $w \rightsquigarrow w'$  (with  $W$  implicit), if there exists a piecewise linear path in  $W$  such that  
 119  $\rho(0) = w$  and  $\rho(1) = w'$ . Note that  $\rightsquigarrow$  is an equivalence relation on  $W$ . A set  $W \subseteq \mathcal{W}_h$  is itself  
 120 *piecewise-linear path-connected* if and only if  $\rightsquigarrow$  is full, that is, all pairs of parameters in  $W$  are  
 121 piecewise linear path-connected in  $W$ .

122 The *length* of a piecewise linear path is the number of maximal linear segments comprising the path.  
 123 The *distance* between two piecewise linear path-connected parameters is the length of the shortest  
 124 path connecting them. The *diameter* of a piecewise linear path-connected set is the largest distance  
 125 between any two parameters in the set.

## 126 4 Parameter Canonicalisation

127 A parameter *canonicalisation algorithm* maps each parameter in a functional equivalence class to a  
 128 canonical representative parameter within that class. A canonicalisation algorithm therefore serves  
 129 as a computational test of functional equivalence.

130 Prior work has described canonicalisation algorithms for certain irreducible parameters (Rüger and  
 131 Ossén, 1997); but when applied to functionally equivalent reducible parameters, such algorithms  
 132 may fail to produce the same output. We introduce a canonicalisation algorithm that properly canon-  
 133 icalises both reducible and irreducible parameters, based on similar negation and sorting stages,  
 134 combined with a novel *reduction* stage. This stage effectively removes or ‘zeroes out’ redundant  
 135 units through various operations, isolating a functionally equivalent but irreducible subparameter.

136 **Algorithm 4.1** (Parameter canonicalisation). Given a parameter space  $\mathcal{W}_h$ , proceed:

```

137 1: procedure CANONICALISE( $w = (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h$ )
138 2:    $\triangleright$  Stage 1: Reduce the parameter, zeroing out redundant hidden units  $\triangleleft$ 
139 3:    $Z \leftarrow \{\}$   $\triangleright$  keep track of ‘zeroed’ units
140 4:   while any of the following four conditions hold do
141 5:     if for some hidden unit  $i \notin Z, a_i = 0$  then  $\triangleright$  reducibility condition (i)
142 6:        $b_i, c_i \leftarrow 0$ 
143 7:        $Z \leftarrow Z \cup \{i\}$ 
144 8:     else if for some hidden unit  $i \notin Z, b_i = 0$  then  $\triangleright$  — (ii)
145 9:        $d \leftarrow d + a_i \tanh(c_i)$ 
146 10:       $a_i, c_i \leftarrow 0$ 
147 11:       $Z \leftarrow Z \cup \{i\}$ 
148 12:     else if for some hidden units  $i, j \notin Z, i \neq j, (b_i, c_i) = (b_j, c_j)$  then  $\triangleright$  — (iii)
149 13:        $a_j \leftarrow a_j + a_i$ 
150 14:        $a_i, b_i, c_i \leftarrow 0$ 
151 15:        $Z \leftarrow Z \cup \{i\}$ 
152 16:     else if for some hidden units  $i, j \notin Z, i \neq j, (b_i, c_i) = (-b_j, -c_j)$  then  $\triangleright$  — (iv)
153 17:        $a_j \leftarrow a_j - a_i$ 
154 18:        $a_i, b_i, c_i \leftarrow 0$ 
155 19:        $Z \leftarrow Z \cup \{i\}$ 
156 20:     end if
157 21:   end while
158 22:    $\triangleright$  Stage 2: Negate the nonzero units to have positive incoming weights  $\triangleleft$ 
159 23:   for each hidden unit  $i \notin Z$  do
160 24:      $a_i, b_i, c_i \leftarrow \text{sign}(b_i) \cdot (a_i, b_i, c_i)$ 
161 25:   end for
162 26:    $\triangleright$  Stage 3: Sort the units by their incoming weights and biases  $\triangleleft$ 
163 27:    $\pi \leftarrow$  a permutation sorting  $i = 1, \dots, h$  by decreasing  $b_i$ , breaking ties with decreasing  $c_i$ 
164 28:    $w \leftarrow (a_{\pi(1)}, b_{\pi(1)}, c_{\pi(1)}, \dots, a_{\pi(h)}, b_{\pi(h)}, c_{\pi(h)}, d)$ 
165 29:    $\triangleright$  Now,  $w$  has been mutated into the canonical equivalent parameter  $\triangleleft$ 
166 30:   return  $w$ 
167 31: end procedure

```

168 The following theorem establishes the correctness of Algorithm 4.1.

169 **Theorem 4.2.** Let  $w, w' \in \mathcal{W}_h$ . Let  $v = \text{CANONICALISE}(w)$  and  $v' = \text{CANONICALISE}(w')$ . Then

- 170 (i)  $v$  is functionally equivalent to  $w$ ; and  
 171 (ii) if  $w$  and  $w'$  are functionally equivalent, then  $v = v'$ .

172 *Proof.* For (i), observe that  $f_w$  is maintained by each iteration of the loops in Stages 1 and 2, and  
 173 by the permutation in Stage 3. For (ii), observe that Stage 1 isolates functionally equivalent *and*  
 174 *irreducible* subparameters  $u \in \mathcal{W}_r$  and  $u' \in \mathcal{W}_{r'}$  of the input parameters  $w$  and  $w'$  (excluding the  
 175 zeroed units). We have  $f_u = f_w = f_{w'} = f_{u'}$ , so by the results of Sussmann (1992),  $r = r' =$   
 176  $\text{rank}(w)$ , and  $u$  and  $u'$  are related by unit negation and exchange transformations. This remains true  
 177 in the presence of the zero units. Stages 2 and 3 are invariant to precisely such transformations by  
 178 construction.  $\square$

## 5 Full Functional Equivalence Class

Algorithm 4.1 produces a consistent output for all parameters within a given functional equivalence class. It serves as the basis for the following characterisation of the full functional equivalence class.

The idea behind the characterisation is to enumerate the various ways for a parameter's units to be reduced, negated, and sorted throughout Algorithm 4.1. Each such *canonicalisation trace* corresponds to a simple set of parameters that takes exactly this path through the algorithm, as follows.

**Definition 5.1** (Canonicalisation trace). Let  $r, h \in \mathbb{N}$ ,  $r \leq h$ . A *canonicalisation trace of order  $r$  on  $h$  units* is a tuple  $(\sigma, \tau)$ , where  $\sigma \in \{-1, +1\}^h$  is a sign vector (interpreted as tracking unit negation throughout the algorithm); and  $\tau : \{1, \dots, h\} \rightarrow \{0, 1, \dots, h\}$  is a function with range including  $\{1, \dots, r\}$  (interpreted as tracking unit reduction and permutation throughout the algorithm).

**Theorem 5.2.** Let  $w \in \mathcal{W}_h$  and  $v = (\alpha_1, \beta_1, \gamma_1, \dots, \alpha_h, \beta_h, \gamma_h, \delta) = \text{CANONICALISE}(w)$ . Let  $r = \text{rank}(w)$ . Then the functional equivalence class  $\mathfrak{F}[w] \subset \mathcal{W}_h$  is a union of subsets

$$\mathfrak{F}[w] = \bigcup_{(\sigma, \tau) \in \Gamma(h, r)} \left( X_{\tau^{-1}[0]}^\delta \cap \bigcap_{i=1}^r Y_{\sigma, \tau^{-1}[i]}^{\alpha_i, \beta_i, \gamma_i} \cap \bigcap_{i=r+1}^h Z_{\sigma, \tau^{-1}[i]} \right) \quad (1)$$

where  $\Gamma(h, r)$  denotes the set of all canonicalisation traces of order  $r$  on  $h$  units and

$$\begin{aligned} X_I^\delta &= \left\{ (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h \mid \begin{array}{l} \forall i \in I, b_i = 0 \text{ and} \\ d + \sum_{i \in I} a_i \tanh(c_i) = \delta \end{array} \right\}; \\ Y_{\sigma, I}^{\alpha, \beta, \gamma} &= \left\{ (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h \mid \begin{array}{l} \forall i \in I, \sigma_i \cdot (b_i, c_i) = (\beta, \gamma) \\ \text{and } \sum_{i \in I} \sigma_i a_i = \alpha \end{array} \right\}; \text{ and} \\ Z_{\sigma, I} &= \left\{ (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h \mid \begin{array}{l} \forall i, j \in I, \sigma_i \cdot (b_i, c_i) = \sigma_j \cdot (b_j, c_j) \\ \text{and } \sum_{i \in I} \sigma_i a_i = 0 \end{array} \right\}. \end{aligned}$$

*Proof.* Suppose  $w' = (a'_1, b'_1, c'_1, \dots, a'_h, b'_h, c'_h, d) \in \mathcal{W}_h$  is in the union in (1), and therefore in the intersection for some canonicalisation trace  $(\sigma, \tau) \in \Gamma(h, r)$ . Then  $f_{w'} = f_v = f_w$ , as follows:

$$\begin{aligned} f_{w'}(x) &= d' + \sum_{i \in \tau^{-1}[0]} a'_i \tanh(b'_i x + c'_i) + \sum_{j=1}^r \sum_{i \in \tau^{-1}[j]} a'_i \tanh(b'_i x + c'_i) + \sum_{j=r+1}^h \sum_{i \in \tau^{-1}[j]} a'_i \tanh(b'_i x + c'_i) \\ &= \delta + \sum_{j=1}^r \alpha_j \tanh(\beta_j x + \gamma_j) \text{ since } w' \in X_{\tau^{-1}[0]}^\delta \cap \bigcap_{j=1}^r Y_{\sigma, \tau^{-1}[j]}^{\alpha_j, \beta_j, \gamma_j} \cap \bigcap_{j=r+1}^h Z_{\sigma, \tau^{-1}[j]}. \end{aligned}$$

Now, suppose  $w' \in \mathfrak{F}[w]$ . Construct a canonicalisation trace  $(\sigma, \tau) \in \Gamma(h, r)$  following the execution of Algorithm 4.1 on  $w'$ . Set  $\sigma_i = -1$  where  $\text{sign}(b'_i) = -1$ , otherwise  $+1$ . Construct  $\tau$  from identity as follows. In each Stage 1 iteration, if the second branch is chosen, remap  $\tau(i)$  to 0. If the third or fourth branch is chosen, for  $k \in \tau^{-1}[i]$  (including  $i$  itself), remap  $\tau(k)$  to  $j$ . Finally, incorporate the Stage 3 permutation  $\pi$ : simultaneously for  $k \notin \tau^{-1}[0]$ , remap  $\tau(k)$  to  $\pi(\tau(k))$ .

Note  $\text{CANONICALISE}(w') = v$  by Theorem 4.2. Then  $w' \in X_{\tau^{-1}[0]}^\delta$  because  $\tau^{-1}[0]$  contains exactly those units incorporated into  $\delta$ . Moreover, for  $j = 1, \dots, r$ ,  $w' \in Y_{\sigma, \tau^{-1}[j]}^{\alpha_j, \beta_j, \gamma_j}$ , because  $\tau^{-1}[j]$  contains exactly those units incorporated into unit  $j$  of  $v$ , and  $\sigma$  their relative signs ( $\beta_j > 0$ ). Likewise, for  $j \in r+1, \dots, h$ ,  $w' \in Z_{\sigma, \tau^{-1}[j]}$  (which is vacuous if  $\tau^{-1}[j]$  is empty).  $\square$

**Remark 5.3.** If  $w \in \mathcal{W}_h$  is irreducible, then  $\text{rank}(w) = h$ . For  $(\sigma, \tau) \in \Gamma(h, h)$ ,  $\tau$  is a permutation (since the range must include  $\{1, \dots, h\}$ ). The set of traces therefore corresponds to the set of transformations generated by unit negations and transpositions, as in Sussmann (1992).

**Remark 5.4.** When  $\text{rank}(w) = h - 1$ , there are, modulo sign vectors and permutations, essentially three canonicalisation traces, corresponding to the three ways of adding an additional unit to a  $(h - 1)$ -unit network discussed by Fukumizu and Amari (2000) and Fukumizu et al. (2019): to introduce a new constant unit or one with zero output, or to split an existing unit in two.

**Remark 5.5.** Similarly, in Şimşek et al. (2021, Definitions 3.2 and 3.3), an  $(r + j)$ -tuple coupled with a permutation play the role of  $\tau$  in characterising the *expansion manifold*, akin to the functional equivalence class but from the dual perspective of adding units to an irreducible parameter. Şimşek et al. (2021) study a setting without a unit negation symmetry, so there is no need for a sign vector.

## 6 Path Connectivity

In this section, we show that the reducible functional equivalence class is piecewise linear path-connected (Theorem 6.1), and, for parameters with rank at most half of the available number of hidden units, has diameter at most 7 linear segments (Theorem 6.3).

**Theorem 6.1.** *Let  $w \in \mathcal{W}_h$ . If  $w$  is reducible, then  $\mathfrak{F}[w]$  is piecewise linear path-connected.*

*Proof.* It suffices to show that each reducible parameter  $w \in \mathcal{W}_h$  is piecewise linear path-connected in  $\mathfrak{F}[w]$  to its canonical representative  $\text{CANONICALISE}(w)$ . The path construction proceeds by tracing the parameter's mutations in the course of execution of Algorithm 4.1. For each iteration of the loops in Stages 1 and 2, and for each transposition in the permutation in Stage 3, we construct a multi-segment sub-path. To describe these sub-paths, we denote the parameter at the beginning of each sub-path as  $w = (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d)$ , noting that this parameter is mutated throughout the algorithm, but is functionally equivalent to the original  $w$  at all of these intermediate points.

1. In each iteration of the Stage 1 loop, the construction depends on the chosen branch, as follows. Some examples are illustrated in Figure 1.

- (i) A direct path interpolating  $b_i$  and  $c_i$  to zero.
- (ii) A two-segment path, interpolating  $a_i$  to zero and  $d$  to  $d + a_i \tanh(c_i)$ , then  $c_i$  to zero.
- (iii) A two-segment path, interpolating  $a_i$  to zero and  $a_j$  to  $a_j + a_i$ , then  $b_i$  and  $c_i$  to zero.
- (iv) A two-segment path, interpolating  $a_i$  to zero and  $a_j$  to  $a_j - a_i$ , then  $b_i$  and  $c_i$  to zero.

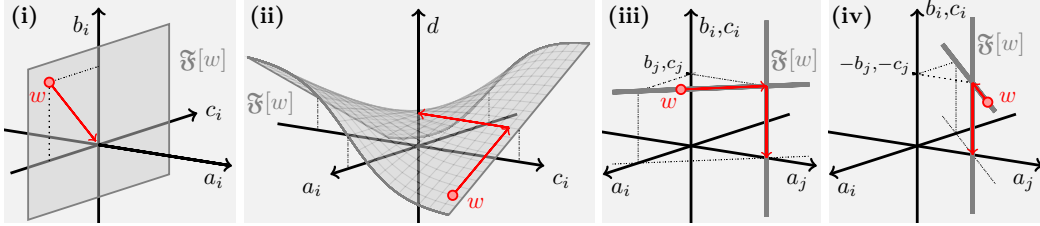


Figure 1: Example paths constructed for each of the Stage 1 branches. Other dimensions held fixed.

Since (the original)  $w$  is reducible, (the current)  $w$  must have gone through at least one iteration in Stage 1, and must have at least one *blank* unit  $k$  with  $a_k, b_k, c_k = 0$ . From any such parameter  $w$ , there is a three-segment path in  $\mathfrak{F}[w]$  that implements a *blank-exchange manoeuvre* transferring the weights of another unit  $i$  to unit  $k$ , and leaving  $a_i, b_i, c_i = 0$ : first interpolate  $b_k$  to  $b_i$  and  $c_k$  to  $c_i$ ; then interpolate  $a_k$  to  $a_i$  and  $a_i$  to zero; then interpolate  $b_i$  and  $c_i$  to zero. Likewise, there is a three-segment path that implements a *negative blank-exchange manoeuvre*, negating the weights as they are interpolated into the blank unit. With these manoeuvres noted, proceed:

2. In each iteration of the Stage 2 loop for which  $\text{sign}(b_i) = -1$ , let  $k$  be a blank unit, and construct a six-segment path. First, blank-exchange unit  $i$  into unit  $k$ . Then, negative blank-exchange unit  $k$  into unit  $i$ . The net effect is to negate unit  $i$ .
3. In Stage 3, construct a path for each segment in a decomposition of the permutation  $\pi$  as a product of transpositions. Consider the transposition  $(i, j)$ . If  $i$  or  $j$  is blank, simply blank-exchange them. If neither is blank, let  $k$  be a blank unit. Construct a nine-segment path, using three blank-exchange manoeuvres, using  $k$  as ‘temporary storage’ to implement the transposition: first blank-exchange units  $i$  and  $k$ , then blank-exchange units  $i$  (now blank) and  $j$ , then blank-exchange units  $j$  (now blank) and  $k$  (containing  $i$ ’s original weights).

The resulting parameter is the canonical representative and it can be verified that each segment in each sub-path remains in  $\mathfrak{F}[w]$  as required.  $\square$

**Remark 6.2.** [Şimşek et al. \(2021, Theorem B.4\)](#) construct similar paths to show the connectivity of their expansion manifold (cf. Remark 5.5). They first connect reduced-form parameters using blank-exchange manoeuvres and then show inductively that each unit addition preserves connectivity.



253 **Theorem 6.3.** Let  $w \in \mathcal{W}_h$ . If  $\text{rank}(w) \leq \frac{h}{2}$ , then  $\mathfrak{F}[w]$  has diameter at most 7.

254 *Proof.* Let  $w \in \mathcal{W}_h$  with  $\text{rank}(w) = r \leq \frac{h}{2}$ . Let  $w' \in \mathfrak{F}[w]$ . We construct a piecewise linear  
 255 path from  $w$  to  $w'$  with 7 segments. By Theorem 6.1, a path exists via the canonical representative  
 256 parameter  $v = \text{CANONICALISE}(w)$ . However, this path has excessive length. We compress the  
 257 length to 7 by exploiting the following opportunities to parallelise segments and ‘cut corners’. These  
 258 optimisation steps are illustrated in Figure 2.

- 259 (a) Let the Stage 1 result from Algorithm 4.1 for  $w$  be denoted  $u$ . Let the Stage 1 result for  $w'$   
 260 be denoted  $u'$ . Instead of following the unit negation and exchange transformations from  $u$   
 261 to  $v$ , and then back to  $u'$ , we transform  $u$  into  $u'$  directly, not (necessarily) via  $v$ .
- 262 (b) We connect  $w$  to  $u$  using two segments, implementing all iterations of Stage 1 in parallel.  
 263 The first segment shifts the outgoing weights from the blank units to the non-blank units and  
 264 the output unit bias. The second segment interpolates the blank units’ incoming weights  
 265 and biases to zero. We apply the same optimisation to connect  $w$  and  $u'$ .
- 266 (c) We connect  $u$  and  $u'$  using two blank-exchange manoeuvres (6 segments), exploiting the  
 267 majority of blank units as ‘temporary storage’. First, we blank-exchange the non-blank  
 268 units of  $u$  into blank units of  $u'$ , resulting in a parameter  $\bar{u}'$  sharing no non-blank units with  
 269  $u'$ . Then, we (negative) blank-exchange those weights into the appropriate non-blank units  
 270 of  $u'$ , implementing the unit negation and exchange transformations relating  $u$ ,  $\bar{u}'$ , and  $u'$ .
- 271 (d) The manoeuvres in (b) and (c) begin and/or end by interpolating incoming weights and  
 272 biases of blank units from and/or to zero, while the outgoing weights are zero. We  
 273 combine adjacent beginning/end segments together, without (necessarily) passing through  
 274 zero. This results in the required seven-segment path, tracing the sequence of parameters  
 275  $w, w^1, w^2, \dots, w^6, w' \in \mathcal{W}_h$ .

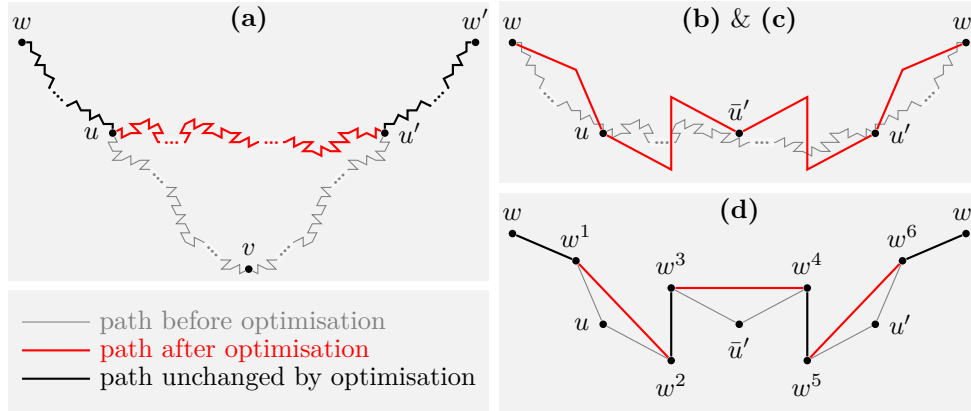


Figure 2: A conceptual illustration of the four path optimisations, producing a seven-segment piecewise linear path of equivalent parameters in a high-dimensional parameter space. (a) Follow unit negation and exchange transformations directly between reduced parameters, not via the canonical parameter. (b) & (c) Parallelise the reduction steps, and use the majority of blank units to parallelise the transformations. (d) Combine first/last segments of reduction and blank-exchange manoeuvres.

276 To describe the constructed path in detail, we introduce the following notation for the components  
 277 of the key parameters  $w, w', u, u', w^1, w^2, \dots, w^6 \in \mathcal{W}_h$ :

$$\begin{aligned} w &= (a_1^w, b_1^w, c_1^w, \dots, a_h^w, b_h^w, c_h^w, d^w) & u &= (a_1^u, b_1^u, c_1^u, \dots, a_h^u, b_h^u, c_h^u, d^u) \\ w' &= (a_1^{w'}, b_1^{w'}, c_1^{w'}, \dots, a_h^{w'}, b_h^{w'}, c_h^{w'}, d^{w'}) & u' &= (a_1^{u'}, b_1^{u'}, c_1^{u'}, \dots, a_h^{u'}, b_h^{u'}, c_h^{u'}, d^{u'}) \\ w^k &= (a_1^k, b_1^k, c_1^k, \dots, a_h^k, b_h^k, c_h^k, d^k) & (k &= 1, \dots, 6). \end{aligned}$$

278 Of the  $h$  units in  $u$ , exactly  $h - r$  are blank—those in the set  $Z$  from  $\text{CANONICALISE}(w)$ . Denote  
 279 the complement set of  $r$  non-blank units  $U = \{1, \dots, h\} \setminus Z$ . Likewise, define  $Z'$  and  $U'$  from  $u'$ .

280 With notation clarified, we can now describe the key points  $w^1, \dots, w^6$  in detail, while showing that  
 281 the entire path is contained within the functional equivalence class  $\mathfrak{F}[w]$ .

282 1. The first segment interpolates each outgoing weight from  $a_i^w$  to  $a_i^u$ , and interpolates the  
 283 output bias from  $d^w$  to  $d^u$ . That is,  $w^1 = (a_1^u, b_1^w, c_1^w, \dots, a_h^u, b_h^w, c_h^w, d^u)$ .

284 To see that this segment is within  $\mathfrak{F}[w]$ , observe that since the incoming weights and biases  
 285 are unchanged between the two parameters,  $f_{tw^1+(1-t)w}(x) = tf_{w^1}(x) + (1-t)f_w(x)$  for  
 286  $x \in \mathbb{R}$  and  $t \in [0, 1]$ . To show that  $f_w = f_{w^1}$ , we construct a function  $\tau : \{1, \dots, h\} \rightarrow$   
 287  $\{0, 1, \dots, h\}$  from identity following each iteration of Stage 1 of CANONICALISE( $w$ ):  
 288 when the second branch is chosen, remap  $\tau(i)$  to 0; and when the third or fourth branch is  
 289 chosen, for  $k \in \tau^{-1}[i]$  (including  $i$  itself), remap  $\tau(k)$  to  $j$ . Moreover, we define a sign  
 290 vector  $\sigma \in \{-1, +1\}^h$  where  $\sigma_i = -1$  if  $\text{sign}(b_i^w) = -1$ , otherwise  $\sigma_i = +1$ . Then:

$$\begin{aligned} f_w(x) &= d^w + \sum_{j=0}^k \sum_{i \in \tau^{-1}[j]} a_i^w \tanh(b_i^w x + c_i^w) \\ &= d^w + \sum_{i \in \tau^{-1}[0]} a_i^w \tanh(c_i^w) + \sum_{j=1}^h \left( \sum_{i \in \tau^{-1}[j]} \sigma_j \sigma_i a_i^w \right) \tanh(b_j^w x + c_j^w) \\ &= d^u + \sum_{j=1}^h a_j^u \tanh(b_j^w x + c_j^w) = f_{w^1}(x). \end{aligned}$$

291 2. The second segment completes the reduction and begins the first blank-exchange ma-  
 292 noeuvre to store the nonzero units in  $Z'$ . For  $i \in U \cap U'$ , pick distinct ‘storage’ units  
 293  $j \in Z \cap Z'$ . There are enough, as  $r \leq \frac{h}{2}$  by assumption thus  $|U \cap U'| = |U| - |Z \cap U'| =$   
 294  $r - |Z \cap U'| \leq (h - r) - |Z \cap U'| = |Z'| - |Z \cap U'| = |Z' \cap Z|$ . Interpolate unit  $j$ ’s  
 295 incoming weight from  $b_j^w$  to  $b_j^u$  and interpolate its bias from  $c_j^w$  to  $c_j^u$ . Meanwhile, for all  
 296 other  $j \in Z$ , interpolate the incoming weight and bias to zero. This segment is within  $\mathfrak{F}[w]$   
 297 as for  $j \in Z$ ,  $a_j^1 = a_j^u = 0$  by definition of  $Z$ .

298 3. The third segment shifts the outgoing weights from the units in  $U \cap U'$  to the units in  $Z \cap Z'$   
 299 prepared in step (2). For  $i \in U \cap U'$ , pick the same storage unit  $j$  as in step (2). Interpolate  
 300 unit  $j$ ’s outgoing weight from  $a_j^w = 0$  to  $a_j^u$  and interpolate unit  $i$ ’s outgoing weight from  
 301  $a_i^u$  to zero. This segment is within  $\mathfrak{F}[w]$  as  $b_i^2 = b_i^2$  and  $c_i^2 = c_i^2$  by step (2).

302 4. The fourth segment completes the first blank-exchange manoeuvre and begins the second,  
 303 to form the units of  $u'$ . For  $i \in U'$ , interpolate unit  $i$ ’s incoming weight from  $b_i^2$  to  $b_i^{u'}$  and  
 304 interpolate its bias from  $c_i^2$  to  $c_i^{u'}$ . This segment is within  $\mathfrak{F}[w]$  because for  $i \in U' \cap Z$ ,  
 305  $a_i^3 = a_i^u = 0$  by definition of  $Z$ , and for  $i \in U' \cap U$ ,  $a_i^3 = 0$  by step (3).

306 5. The fifth segment shifts the outgoing weights from the selected units in  $Z'$  to the units in  
 307  $U'$  prepared in step (4). We simply interpolate each unit  $i$ ’s outgoing weight to  $a_i^{u'}$ .

308 To see that the segment is within  $\mathfrak{F}[w]$ , note that  $u$  and  $u'$  are related by some unit negation  
 309 and exchange transformations. Therefore, there is a correspondence between their sets of  
 310 nonzero units, such that corresponding units have the same (or negated) incoming weights  
 311 and biases. Due to steps (2)–(4) there are  $r$  ‘storage’ units in  $w^4$  with the weights of the  
 312 units of  $u$ , and the correspondence extends to these storage units. Since the storage units  
 313 are disjoint with  $U'$ , this fifth segment has the effect of interpolating the outgoing weight  
 314 of each of the storage units  $j \in Z'$  in  $w^4$  from  $a_j^u$  to zero (where  $i$  is as in step (3)), while  
 315 interpolating the outgoing weight of its corresponding unit  $k \in U'$  from zero to  $\pm a_k^u = a_k^{u'}$   
 316 (where the sign depends on the unit negation transformations relating  $u$  and  $u'$ ).

317 6. The sixth segment completes the second blank-exchange manoeuvre and begins to reverse  
 318 the reduction. For  $i \in Z'$ , interpolate unit  $i$ ’s incoming weight from  $b_i^5$  to  $b_i^{w'}$ , and interpo-  
 319 late its bias from  $c_i^5$  to  $c_i^{w'}$ . This segment is within  $\mathfrak{F}[w]$  as for  $i \in Z'$ ,  $a_i^5 = a_i^{u'} = 0$  by  
 320 definition of  $Z'$ .

321 7. The seventh segment, of course, interpolates from  $w^6$  to  $w'$ . To see that this segment is  
 322 within  $\mathfrak{F}[w]$ , note that by steps (5) and (6),  $w^6 = (a_1^{u'}, b_1^{w'}, c_1^{w'}, \dots, a_h^{u'}, b_h^{w'}, c_h^{w'}, d^{u'})$   
 323 (noting  $d^u = d^{u'}$  since the output unit’s bias is preserved by unit transformations). So the  
 324 situation is the reverse of step (1), and a similar proof applies.  $\square$



## 7 Discussion

In this paper, we have investigated the functional equivalence class for reducible neural network parameters, and its connectivity properties. These reducible functional equivalence classes are a complex union of manifolds, displaying the following rich qualitative structure.

- There is a central discrete array of reduced-form parameters, with a maximal number of blank units spread throughout an irreducible subnetwork. These reduced-form parameters are related by unit negation and exchange transformations, like for irreducible parameters.
- Unlike in the irreducible case, these reduced-form parameters are connected by a network of piecewise linear paths. Namely, these are (negative) blank-exchange manoeuvres, and, when there are multiple blank units, simultaneous parallel blank-exchange manoeuvres.
- Various manifolds branch away from this central network, tracing in reverse the various reduction operations (optionally in parallel). Dually, these manifolds trace methods for *adding* units (cf., [Fukumizu and Amari, 2000](#); [Fukumizu et al., 2019](#); [Şimşek et al., 2021](#)).

Theorem 6.3 establishes that with a majority of blank units, the diameter of this parameter network becomes a small constant number of linear segments. With fewer blank units it will sometimes require more blank-exchange manoeuvres to traverse the central network. Future work could investigate the trade-offs between shortest path length and rank for different unit permutations.

**Towards modern architectures.** We have studied single-hidden-layer hyperbolic tangent networks, but structural redundancies arising from zero, constant, or proportional units (reducibility conditions (i)–(iii)) are a generic feature of feed-forward network components. Unit negation symmetries are characteristic of odd nonlinearities; other nonlinearities will exhibit similar redundancies due to their own affine symmetries. In more complex architectures there will be additional sources of redundancy, such as interactions between layers or specialised computational structures.

We call for future work to seek out, catalogue, and thoroughly investigate such sources of redundancy, rather than assuming their irrelevance as part of measure zero subset of the parameter space. Our results serve as a starting point for future work in this direction. The results of [Vlačić and Bölcskei \(2021\)](#), significantly generalising [Sussmann \(1992\)](#), would be a useful complement.

**Functional equivalence and deep learning.** Functionally equivalent parameters have equal loss. Continuous directions and piecewise linear paths within reducible functional equivalence classes (Theorems 5.2, 6.1, and 6.3) therefore imply flat directions and equal-loss paths in the loss landscape. More broadly, the set of low- or zero-loss parameters is a union of functional equivalence classes, including, possibly (or *necessarily*, given sufficient overparameterisation), reducible ones.

Understanding reducible functional equivalence classes may be key to understanding these topics. Of special interest is the connection to theoretical work involving unit pruning ([Kuditipudi et al., 2019](#)) and permutation symmetries ([Brea et al., 2019](#)). Of course, having the same loss does not imply functional equivalence—indeed, [Garipov et al. \(2018\)](#) observe functional non-equivalence in low-loss paths. The exact relevance of reducible parameters to these topics remains to be clarified.

If the loss landscape is smooth, the comments above hold approximately for irreducible parameters that are merely near some reducible parameter. Future work should develop techniques to measure proximity to low-rank parameters, and empirically investigate the prevalence of approximate reducibility among parameters encountered during learning.

## 8 Conclusion

While reducible parameters comprise a measure zero subset of the parameter space, their functional equivalence classes may still be key to understanding the structure of the parameter space and, in turn, the loss landscape on which deep learning takes place. We have taken the first step towards understanding functional equivalence beyond irreducible parameters, by investigating the setting of single-hidden-layer hyperbolic tangent networks. Due to structural redundancy, reducible functional equivalence classes are much richer than their irreducible counterparts. By accounting for various kinds of structural redundancy, we offer a characterisation of reducible functional equivalence classes and an investigation of their piecewise linear connectivity properties.

## References

- Francesca Albertini and Eduardo D. Sontag. For neural networks, function determines form. Technical Report SYCON-92-03, Rutgers Center for Systems and Control, **1992**. Expanded version of [Albertini and Sontag \(1993a\)](#). Cited on page 2.
- Francesca Albertini and Eduardo D. Sontag. For neural networks, function determines form. *Neural Networks*, 6(7):975–990, **1993a**. Access via [Crossref](#). Cited on pages 2 and 10.
- Francesca Albertini and Eduardo D. Sontag. Identifiability of discrete-time neural networks. In *Proceedings of the European Control Conference 1993*, volume 2, pages 460–465. European Control Association, **1993b**. Access via [Francesca Albertini](#). Cited on page 2.
- Francesca Albertini and Eduardo D. Sontag. Uniqueness of weights for recurrent nets. In *Systems and Networks: Mathematical Theory and Applications: Proceedings of the International Symposium MTNS 1993*, volume II, pages 599–602. Akademie Verlag, **1993c**. Access via [Francesca Albertini](#) or via [Eduardo D. Sontag](#). See also extended version, access via [Eduardo D. Sontag](#). Cited on page 2.
- Francesca Albertini, Eduardo D. Sontag, and Vincent Maillot. Uniqueness of weights for neural networks. In *Artificial Neural Networks for Speech and Vision*, pages 113–125. Chapman & Hall, London, **1993**. Proceedings of a workshop held at Rutgers University in 1992. Access via [Eduardo D. Sontag](#). Cited on pages 1 and 2.
- Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. Parameter identifiability of a deep feedforward ReLU neural network. **2021**. Preprint [arXiv:2112.12982](#) [math.ST]. Cited on page 2.
- Johanni Brea, Berfin Şimşek, Bernd Illing, and Wulfram Gerstner. Weight-space symmetry in deep networks gives rise to permutation saddles, connected by equal-loss valleys across the loss landscape. **2019**. Preprint [arXiv:1907.02911](#) [cs.LG]. Cited on page 9.
- An Mei Chen and Robert Hecht-Nielsen. On the geometry of feedforward neural network weight spaces. In *Second International Conference on Artificial Neural Networks*, pages 1–4. IET, **1991**. Access via [IEEE Xplore](#). Cited on page 2.
- An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. On the geometry of feedforward neural network error surfaces. *Neural Computation*, 5(6):910–927, **1993**. Access via [Crossref](#). Cited on page 2.
- Charles Fefferman. Reconstructing a neural net from its output. *Revista Matemática Iberoamericana*, 10(3):507–555, **1994**. Access via [Crossref](#). Cited on page 2.
- Charles Fefferman and Scott Markel. Recovering a feed-forward net from its output. In *Advances in Neural Information Processing Systems 6*, pages 335–342. Morgan Kaufmann, **1993**. Access via [NeurIPS](#). Cited on page 2.
- Kenji Fukumizu. A regularity condition of the information matrix of a multilayer perceptron network. *Neural Networks*, 9(5):871–879, **1996**. Access via [Crossref](#). Cited on pages 1 and 12.
- Kenji Fukumizu and Shun-ichi Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13(3):317–327, **2000**. Access via [Crossref](#). Cited on pages 2, 5, and 9.
- 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 32*, pages 13868–13876. Curran Associates, **2019**. Access via [NeurIPS](#). Cited on pages 2, 5, and 9.
- Timur Garipov, Pavel Izmailov, Dmitrii Podoprikin, Dmitry P. Vetrov, and Andrew G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems 31*, pages 8789–8798. Curran Associates, **2018**. Access via [NeurIPS](#). Cited on page 9.

- 423 Egbert Harzheim. *Ordered Sets*. Springer, **2005**. Access via [Crossref](#). Cited on page 12.
- 424 Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Ad-*  
425 *vanced Neural Computers*, pages 129–135. North-Holland, Amsterdam, **1990**. Access via [Cross-](#)  
426 [ref](#). Cited on pages 1 and 2.
- 427 Věra Kůrková and Paul C. Kainen. Functionally equivalent feedforward neural networks. *Neural*  
428 *Computation*, 6(3):543–558, **1994**. Access via [Crossref](#). Cited on pages 1 and 2.
- 429 Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev  
430 Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. In *Advances in*  
431 *Neural Information Processing Systems 32*, pages 14601–14610. Curran Associates, **2019**. Access  
432 via [NeurIPS](#). Cited on page 9.
- 433 Mary Phuong and Christoph H. Lampert. Functional vs. parametric equivalence of ReLU networks.  
434 In *8th International Conference on Learning Representations*. OpenReview, **2020**. Access via  
435 [OpenReview](#). Cited on pages 1 and 2.
- 436 Stefan M. Rügger and Arnfried Ossen. The metric structure of weight space. *Neural Processing*  
437 *Letters*, 5(2):1–9, **1997**. Access via [Crossref](#). Cited on pages 2 and 4.
- 438 Berfin Şimşek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerst-  
439 ner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks:  
440 Symmetries and invariances. In *Proceedings of the 38th International Conference on Machine*  
441 *Learning*, pages 9722–9732. PMLR, **2021**. Access via [PMLR](#). Cited on pages 2, 5, 6, and 9.
- 442 Pierre Stock and Rémi Gribonval. An embedding of ReLU networks and an analysis of their identi-  
443 fiability. *Constructive Approximation*, **2022**. Access via [Crossref](#). Cited on page 2.
- 444 Héctor J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-  
445 output map. *Neural Networks*, 5(4):589–593, **1992**. Access via [Crossref](#). Cited on pages 1, 2, 3,  
446 4, 5, 9, 12, 13, 14, and 15.
- 447 Verner Vlačić and Helmut Bölcskei. Affine symmetries and neural network identifiability. *Advances*  
448 *in Mathematics*, 376:107485, **2021**. Access via [Crossref](#). Cited on pages 1, 2, and 9.
- 449 Verner Vlačić and Helmut Bölcskei. Neural network identifiability for a family of sigmoidal non-  
450 linearities. *Constructive Approximation*, 55(1):173–224, **2022**. Access via [Crossref](#). Cited on  
451 page 2.

## A Generalising to multi-dimensional inputs and outputs

In this appendix, we consider a slightly more general family of architectures than that introduced in Section 3. Namely, we consider a family of fully-connected, feed-forward neural network architectures with  $n \in \mathbb{N}^+$  input units,  $m \in \mathbb{N}^+$  biased linear output units, and a single hidden layer of  $h \in \mathbb{N}$  biased hidden units with the hyperbolic tangent nonlinearity. With minor modifications, described in the remainder of this appendix, all definitions, algorithms, theorems, and proofs directly generalise from the case  $n = m = 1$  to arbitrary  $n$  and  $m$ .

**Multi-dimensional architecture.** Let  $n \in \mathbb{N}^+$ ,  $m \in \mathbb{N}^+$ , and  $h \in \mathbb{N}$ . Define the generalised parameter space  $\mathcal{W}_h^{n,m} = \mathbb{R}^{(n+m+1)h+m}$ . The weights and biases of the network’s units are encoded in the parameter vector in the format  $(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) = w \in \mathcal{W}_h^{n,m}$  where for each hidden unit  $i = 1, \dots, h$  there is an *outgoing weight vector*  $a_i \in \mathbb{R}^m$ , an *incoming weight vector*  $b_i \in \mathbb{R}^n$ , and a *bias*  $c_i \in \mathbb{R}$ ; and  $d \in \mathbb{R}^m$  is an *output unit bias vector* containing one bias value for each output unit. This time,  $w$  indexes a multi-dimensional mathematical function  $f_w : \mathbb{R}^n \rightarrow \mathbb{R}^m$  defined as follows:

$$f_w(x) = d + \sum_{i=1}^h a_i \tanh(b_i \cdot x + c_i). \quad (2)$$

Note that we use the same tuple notation and ordering  $(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d)$  but now the  $a_i$ , the  $b_i$ , and  $d$  all denote multi-component vectors. Accordingly, in Equation (2),  $b_i$  and  $x$  are now multiplied using the inner (dot) product, rather than scalar multiplication, since they are both vectors in  $\mathbb{R}^n$ . Moreover,  $a_i \in \mathbb{R}^m$  as a vector is to be multiplied by the scalar  $\tanh(b_i \cdot x + c_i)$ . That is, the sum is over vectors of contributions to output units from each hidden unit.

To generalise the results of the main paper to this setting the first change necessary is to replace all mentions of scalar weights with these vectors of weights, and other similar changes such as reading the literal zero as vector zero where appropriate.

**Signing and sorting incoming weight vectors.** The lexicographic order on  $\mathbb{R}^n$ , denoted  $\preceq$ , is a relation such that for  $u, v \in \mathbb{R}^n$ ,  $u \preceq v$  if and only if  $u = v$  or, in the first index  $i = 1, \dots, n$  where  $u$  and  $v$  differ,  $u_i < v_i$ . From this definition we follow the usual conventions in defining  $\prec$ ,  $\succ$ , and  $\succeq$ . Finally, define the *lexicographic sign* of  $v \in \mathbb{R}^n$ , denoted  $\text{sign}_{\text{lex}}(v)$ , as follows:

$$\text{sign}_{\text{lex}}(v) = \begin{cases} +1 & (v \succ 0), \\ 0 & (v = 0), \\ -1 & (v \prec 0). \end{cases}$$

The parameter canonicalisation algorithm and some of the other theorems and proofs make repeated use of the signs of incoming weight vectors. The lexicographic sign satisfies the requisite properties of the scalar sign function in these uses and so the second change necessary to generalising the results is to replace uses of  $\text{sign}(\cdot)$  with uses of  $\text{sign}_{\text{lex}}(\cdot)$ .

This lexicographic order relation is of course also a total order (see, e.g., Harzheim, 2005, Theorem 4.1.11). Therefore, it allows one to sort a list of vectors. Sorting units by decreasing incoming weights is a key step in Stage 3 of Algorithm 4.1, and so the third change necessary is to use decreasing lexicographic order ( $\succeq$ ) in this stage.

**Generalising Sussmann’s equivalence theorem.** The proofs in the main paper rely on the results of Sussmann (1992) on the equivalence between reducibility and non-minimality, and the fact that irreducible functionally equivalent parameters are related by unit negation and exchange transformations. Sussmann (1992) studied a setting with multiple input units but only a single output unit. Lemmas A.1 and A.2 generalise these results to the multi-output setting.<sup>1</sup> The final necessary change to generalise the results in the main paper is to replace all references to Sussmann’s results with references to Lemma A.1 or Lemma A.2.

<sup>1</sup>The proofs reduce the multi-output case to the single-output case, so they still rely on the results of Sussmann (1992). A generalisation similar to Lemma A.1 is given by Fukumizu (1996).

493 The definitions of unit negation and exchange transformations, reducibility, and non-minimality all  
 494 generalise to arbitrary  $n$  and  $m$  with the above-mentioned changes. These definitions are repeated  
 495 here for convenience.

496 A unit negation transformation is a function  $\sigma_i : \mathcal{W}_h^{n,m} \rightarrow \mathcal{W}_h^{n,m}$  for  $i = 1, \dots, h$ , where

$$\sigma_i(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) = (a_1, b_1, c_1, \dots, -a_i, -b_i, -c_i, \dots, a_h, b_h, c_h, d).$$

497 A unit exchange transformation is a function  $\tau_{i,j} : \mathcal{W}_h^{n,m} \rightarrow \mathcal{W}_h^{n,m}$  for  $i, j = 1, \dots, h$ , where

$$\tau_{i,j}(a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) = (a_1, b_1, c_1, \dots, c_{i-1}, a_j, b_j, c_j, a_{i+1}, \dots, c_{j-1}, a_i, b_i, c_i, a_{j+1}, \dots, a_h, b_h, c_h, d).$$

498 A parameter  $w = (a_1, b_1, c_1, \dots, a_h, b_h, c_h, d) \in \mathcal{W}_h^{n,m}$  is *reducible* if and only if it satisfies any of  
 499 the following conditions (otherwise,  $w$  is *irreducible*):

- 500 (i)  $a_i = 0$  for some  $i$ ,
- 501 (ii)  $b_i = 0$  for some  $i$ ,
- 502 (iii)  $(b_i, c_i) = (b_j, c_j)$  for some  $i \neq j$ , or
- 503 (iv)  $(b_i, c_i) = (-b_j, -c_j)$  for some  $i \neq j$ .

504 A parameter  $w \in \mathcal{W}_h^{n,m}$  is *non-minimal* if and only if  $w$  is functionally equivalent to some  $w' \in$   
 505  $\mathcal{W}_{h'}^{n,m}$  with fewer hidden units  $h' < h$ .

506 **Lemma A.1.** For  $w \in \mathcal{W}_h^{n,m}$ ,  $w$  is reducible if and only if  $w$  is non-minimal.

507 *Proof.* ( $\Rightarrow$ ): A smaller functionally equivalent parameter can be constructed as follows.

- 508 (i) If  $a_i = 0$  for some  $i$ , then hidden unit  $i$  fails to contribute to the function. Construct a  
 509 functionally equivalent parameter  $w' \in \mathcal{W}_{h-1}^{n,m}$  with hidden unit  $i$  omitted:

$$w' = (a_1, b_1, c_1, \dots, a_{i-1}, b_{i-1}, c_{i-1}, a_{i+1}, b_{i+1}, c_{i+1}, \dots, a_h, b_h, c_h, d).$$

- 510 (ii) If  $b_i = 0$  for some  $i$ , then hidden unit  $i$  contributes only a constant to the function. Construct  
 511 a functionally equivalent parameter  $w' \in \mathcal{W}_{h-1}^{n,m}$  with hidden unit  $i$  omitted and the output  
 512 unit bias vector changed to compensate:

$$w' = (a_1, b_1, c_1, \dots, a_{i-1}, b_{i-1}, c_{i-1}, a_{i+1}, b_{i+1}, c_{i+1}, \dots, a_h, b_h, c_h, d + a_i \tanh(c_i)).$$

- 513 (iii) If  $(b_i, c_i) = (b_j, c_j)$  for some  $i \neq j$ , then hidden units  $i$  and  $j$  contribute proportionately.  
 514 They can be combined into a single unit (say,  $j$ ) with the same incoming weights and bias,  
 515 and a combined outgoing weight vector. Construct a functionally equivalent parameter  
 516  $w' \in \mathcal{W}_{h-1}^{n,m}$  accordingly:

$$w' = (a_1, b_1, c_1, \dots, c_{i-1}, a_{i+1}, \dots, c_{j-1}, a_j + a_i, b_j, c_j, a_{j+1}, \dots, a_h, b_h, c_h, d).$$

- 517 (iv) If  $(b_i, c_i) = -(b_j, c_j)$  for some  $i \neq j$ , then hidden units  $i$  and  $j$  contribute in negative  
 518 proportion. Due to the odd property of  $\tanh$  they can be combined into a single unit  
 519 (say,  $j$ ) with incoming weight and bias vectors  $(b_j, c_j)$  and a combined outgoing weight  
 520 vector. Construct a new parameter  $w' \in \mathcal{W}_{h-1}^{n,m}$  accordingly:

$$w' = (a_1, b_1, c_1, \dots, c_{i-1}, a_{i+1}, \dots, c_{j-1}, a_j - a_i, b_j, c_j, a_{j+1}, \dots, a_h, b_h, c_h, d).$$

521 In all cases, the new parameter  $w' \in \mathcal{W}_{h-1}^{n,m}$  has  $f_{w'} = f_w$ , so  $w$  is non-minimal.

522 ( $\Leftarrow$ ): We reduce to the single-output case and apply the result of [Sussmann \(1992\)](#) to show that  $w$   
 523 satisfies at least one of the reducibility conditions.

524 To reduce to the single-output case, we introduce some notation. From the function  $f_w : \mathbb{R}^n \rightarrow \mathbb{R}^m$   
 525 define a series of component functions  $f_w^{(1)}, f_w^{(2)}, \dots, f_w^{(m)} : \mathbb{R}^n \rightarrow \mathbb{R}$  such that for  $x \in \mathbb{R}^n$ ,

$$f_w(x) = (f_w^{(1)}(x), f_w^{(2)}(x), \dots, f_w^{(m)}(x)).$$

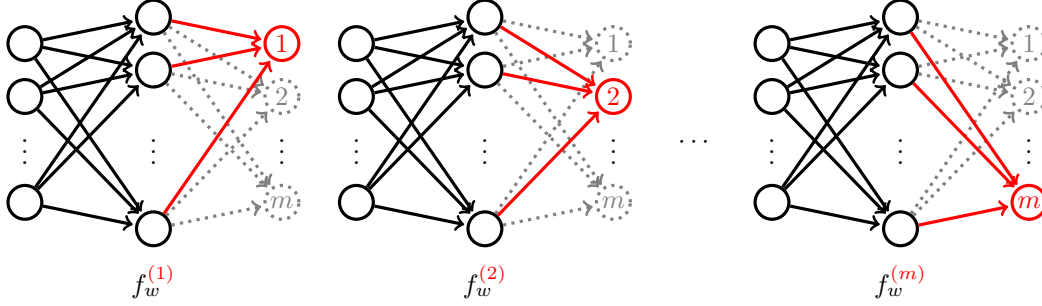


Figure 3: The connection graphs of the component functions of  $f_w$ . Included units and weights are solid. The hidden units of each network share the same incoming weights (and biases, not shown).

Each of these component functions is a simple neural network function in an architecture with  $n$  input units and 1 output unit, corresponding to a subgraph of the connection graph of the original neural network, as illustrated in Figure 3.

Denote the corresponding (overlapping) subvectors of  $w \in \mathcal{W}_h^{n,m}$  as  $w_{(1)}, \dots, w_{(m)} \in \mathcal{W}_h^{n,1}$ . That is, for  $\mu = 1, \dots, m$ ,

$$w_{(\mu)} = (a_{1,\mu}, b_1, c_1, \dots, a_{h,\mu}, b_h, c_h, d_\mu) \in \mathcal{W}_h^{n,1}.$$

Now, let  $w' = (a'_1, b'_1, c'_1, \dots, a'_{h'}, b'_{h'}, c'_{h'}, d') \in \mathcal{W}_{h'}^{n,1}$  such that  $f_{w'} = f_w$  where  $h'$  is the smallest number of hidden units required to implement  $f_w$  ( $h' < h$  by assumption of non-minimality). Apply the same decomposition to  $f_{w'}$  to define  $f_{w'}^{(1)}, \dots, f_{w'}^{(m)}$ , and to define  $w'_{(1)}, \dots, w'_{(m)} \in \mathcal{W}_{h'}^{n,1}$ .

Apply the results of [Sussmann \(1992\)](#) as follows. Since  $f_w = f_{w'}$ ,  $f_w^{(\mu)} = f_{w'}^{(\mu)}$  for  $\mu = 1, \dots, m$ . It follows that for each  $w_{(\mu)}$ ,  $w'_{(\mu)}$  is a functionally equivalent parameter using fewer units. Therefore, the reducibility conditions (in the special case of  $m = 1$ ) must hold for each  $w_{(\mu)}$  ([Sussmann, 1992](#)).

Since conditions (ii–iv) only depend on incoming weights and biases, if any of these conditions hold for any  $w_{(\mu)}$ , then they must also hold for  $w$  itself (which shares the same incoming weights and biases), and the proof is complete. It remains only to consider the case in which conditions (ii–iv) fail to hold for any  $w_{(\mu)}$ , and to show that condition (i) holds for  $w$  itself in this case.

We must introduce yet further notation. For  $i = 1, \dots, h$  denote by  $\varphi_i : \mathbb{R}^n \rightarrow \mathbb{R}$  the function  $\varphi_i(x) = \tanh(b_i x + c_i)$ . Similarly for  $j = 1, \dots, h'$  denote by  $\psi_j : \mathbb{R}^n \rightarrow \mathbb{R}$  the function  $\psi_j(x) = \tanh(b'_j x + c'_j)$ . Then, since we have ruled out reducibility conditions (ii–iv) for  $w$ , no  $\varphi_i$  is constant (ii) and no two are proportional (iii, iv). The same holds for the  $\psi_j$ —conditions (i–iv) do not hold for  $w'_{(\mu)}$ , since  $h'$  was assumed to be minimal. Yet, for  $\mu = 1, \dots, m$ , the linear combination of functions

$$d_\mu + \sum_{i=1}^h a_{i,\mu} \varphi_i - d'_\mu - \sum_{j=1}^{h'} a'_{j,\mu} \psi_j = f_w^{(\mu)} - f_{w'}^{(\mu)} = 0$$

yields the zero function. This linear combination remains when excluding those terms with  $a_{i,\mu} = 0$  or  $a'_{j,\mu} = 0$ . Applying the same reasoning as that in [Sussmann \(1992\)](#), due to the independence property of the hyperbolic tangent function ([Sussmann, 1992](#), Lemma 3.1) the remaining terms must be in bijection, such that

$$\varphi_i = \pm \psi_j \tag{3}$$

for some  $j$  with  $a'_{j,\mu} \neq 0$  for each  $i$  with  $a_{i,\mu} \neq 0$ .

To complete the proof, note that these relationships (3) between the units of  $w$  and  $w'$  are independent of  $\mu$ . However, the relationships are “exclusive” in the sense that no two  $\varphi_i$  can be proportional to the same  $\psi_j$ , else they would also be proportional to each other (ruled out above). Since there are only  $h'$  units  $\psi_1, \dots, \psi_{h'}$ , it follows that there must be one hidden unit  $i$  (actually at least  $h - h'$  many units) for which  $a_{i,\mu} = 0$  for all  $\mu = 1, \dots, m$  (allowing  $\varphi_i$  to avoid any such relationship). That is,  $a_i = (a_{i,1}, \dots, a_{i,m}) = 0$ , satisfying condition (i) for  $w$  as required.  $\square$



558 **Lemma A.2.** *Let  $w \in \mathcal{W}_h^{n,m}$  be irreducible, and let  $w' \in \mathcal{W}_h^{n,m}$ . If  $w$  and  $w'$  are functionally*  
559 *equivalent then there exists a compositional chain of unit negation and exchange transformations,*  
560 *collectively a transformation  $T : \mathcal{W}_h^{n,m} \rightarrow \mathcal{W}_h^{n,m}$ , such that  $w' = T(w)$ .*

561 *Proof.* Once again, we reduce to the case  $m = 1$  and appeal to [Sussmann \(1992\)](#).

562 Suppose  $w' \in \mathfrak{F}[w]$ . Introduce the same decomposition of the two neural networks as in the proof of  
563 Lemma A.1, namely, the component functions  $f_w^{(1)}, \dots, f_w^{(m)}, f_{w'}^{(1)}, \dots, f_{w'}^{(m)}$  implemented by the  
564 parameter subvectors  $w_{(1)}, \dots, w_{(m)}, w'_{(1)}, \dots, w'_{(m)} \in \mathcal{W}_h^{n,1}$  (cf. Figure 3).

565 For  $\mu = 1, \dots, m$ , since  $f_w = f_{w'}$ , we have that  $f_w^{(\mu)} = f_{w'}^{(\mu)}$ . Now,  $w_{(\mu)}$  and  $w'_{(\mu)}$  are not  
566 necessarily irreducible, but if they are reducible then it is only by condition (i), since  $w_{(\mu)}$  and  $w'_{(\mu)}$   
567 have the incoming weights and biases of  $w$  and  $w'$  respectively ( $w$  is irreducible by assumption;  $w'$  is  
568 irreducible because, with the same number of units as  $w$ , it is necessarily minimal, and irreducibility  
569 follows by Lemma A.1). Remove such units with zero outgoing weight from  $w_{(\mu)}$  and  $w'_{(\mu)}$  to  
570 produce new, functionally equivalent irreducible parameters  $u_{(\mu)}, u'_{(\mu)} \in \mathcal{W}_{\text{rank}(w_{(\mu)})}^{n,1}$ . Now by  
571 [Sussmann \(1992, Theorem 2.1\)](#) there exists a chain of unit negation and exchange transformations  
572  $T_\mu$  such that  $u_{(\mu)} = T_\mu(u'_{(\mu)})$ .

573 For each  $\mu$ ,  $T_\mu$  implies a relationship between the units of  $w_{(\mu)}$  and  $w'_{(\mu)}$  with nonzero outgoing  
574 weights, including possible negations and permutations of these units. This same relationship must  
575 hold between those units of  $w$  and  $w'$  since they share incoming weights and biases with  $w_{(\mu)}$   
576 and  $w'_{(\mu)}$ , and (since  $w$  is irreducible, conditions (ii–iv)) these incoming weights are nonzero and  
577 the incoming weight and bias vectors are absolutely distinct between units of the same parameter.  
578 Moreover, all units are involved in some such relationship because no unit of  $w$  or  $w'$  can have zero  
579 outgoing weight vector by reducibility condition (i).

580 So, one can construct from these implied relationships a composition of unit negation and exchange  
581 transformations relating  $w$  and  $w'$  as required.  $\square$