Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Chair for Mathematical Information Science
Prof. Dr. H. Bölcskei

Marco Milanta

Degeneration phenomenon for quantized neural networks

Abstract

Neural networks as a tool to parametrize functions have been widely studied. One important property of a good parametrization is that it cannot be highly redundant. While for some activation functions we know [1] that the neural network's parametrization are almost unique, for ReLU the problem is much harder, and redundant parametrization seem to appear everywhere. In this paper we approach the problem on a slightly different point of view, and we find that such redundancies are actually very rare for deeper networks.

1 Notation

Here we will use $\#A$ to denote the cardinality of the set A.

For the asyntotical behavior we will use the Bachmann–Landau big-O notation.

$$f(n) \in O(g(n)) \Leftrightarrow \exists k > 0 \exists n_0 \forall n > n_0 : |f(n)| \leq k \cdot g(n)$$
$$f(n) \in \Theta(g(n)) \Leftrightarrow \exists k_1 > 0 \exists k_2 > 0 \exists n_0 \forall n > n_0 : k_1 \cdot g(n) \leq f(n) \leq k_2 \cdot g(n)$$
$$f(n) \in \Omega(g(n)) \Leftrightarrow \exists k > 0 \exists n_0 \forall n > n_0 : f(n) \geq k \cdot g(n)$$

1.1 Neural networks

For neural networks we will use the following notation

Definition 1. *(Neural network). We call an ordered sequence*

$$\Phi = (N_0, D_1, \ldots, D_L, W_1, \theta_1, W_2, \theta_2, \ldots, W_L, \theta_L)$$

a neural network, where

- *L is an integer, referred to as the depth of N*
- *$(D_0, \dots, D_L)$ is a $(L+1)$-tuple of integers called layout*
- *$W_\ell \in \mathbb{R}^{D_\ell \times D_{\ell-1}}, \ell \in 1:L$, are matrices so-called weights*
- *$\theta_\ell \in \mathbb{R}^{D_\ell}, \ell \in 1:L$, are vector so-called biases*

Now, we define the main properties of a neural network:

Definition 2. *(Depth, width and connectivity) Given a neural network*

$$\Phi = (N_0, D_1, \dots, D_L, W_1, \theta_1, W_2, \theta_2, \dots, W_L, \theta_L)$$

- *We call depth of Φ*
$$\mathcal{L}(\Phi) := L$$
- *We call width of Φ*
$$\mathcal{W}(\Phi) := \max_{\ell \in 0:L} D_\ell$$
- *We call connectivity of Φ*
$$\mathcal{M}(\Phi) = \sum_{\ell=1}^{L} \sum_{i=1}^{D_\ell} \left(\mathbb{I}_{[\theta_\ell]_i \neq 0} + \sum_{j=1}^{D_{\ell-1}} \mathbb{I}_{[W_\ell]_{i,j} \neq 0} \right)$$

  Which is simply the number of non-zero parameters.

To move from the space of neural networks to the space of functions we use the concept of realization.

Definition 3. *(Neural network realization). Given a function $\rho : \mathbb{R} \to \mathbb{R}$, referred to as activation function, and a neural network Φ we define a map $\langle \Phi \rangle^\rho =: \mathbb{R}^{D_0} \to \mathbb{R}^{D_L}$ given by*

$$\langle \Phi \rangle^\rho = A_L \circ \rho \circ A_{L-1} \circ \rho \circ \cdots \circ A_2 \circ \rho \circ A_1.$$

Where $A_\ell(x) = W_\ell x + \theta_\ell, \ell \in 1:L$. And where ρ acts on vector in a component wise fashion.

2 Introduction

When we think about neural networks as a way to parametrize a high variety of functions we would like to make sure that each different neural network realizes a different function. If we think about it, however, it's trivial that just swapping nodes will not make any difference, hence the parametrization is redundant. But not all hope is lost: [1] found that, for some activation functions, excluding permutation of nodes within the same layer, we can make sure that our function parametrization is unique. But the result doesn't hold for ReLU.

ReLU networks are much harder to cope with. It is easy to find neural networks with different depth, width and weights which still realize the same function.

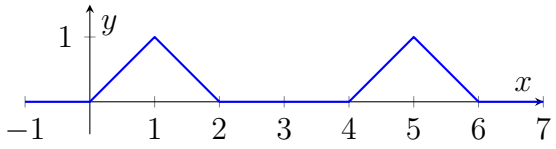


Figure 1: f function

For instance one can notice that any ReLU network, can also be realized by a depth 2 ReLU network. This is because any ReLU network will realize a continuous piecewise-linear function, and such functions, can always be realized by ReLU neural networks with one hidden layer.

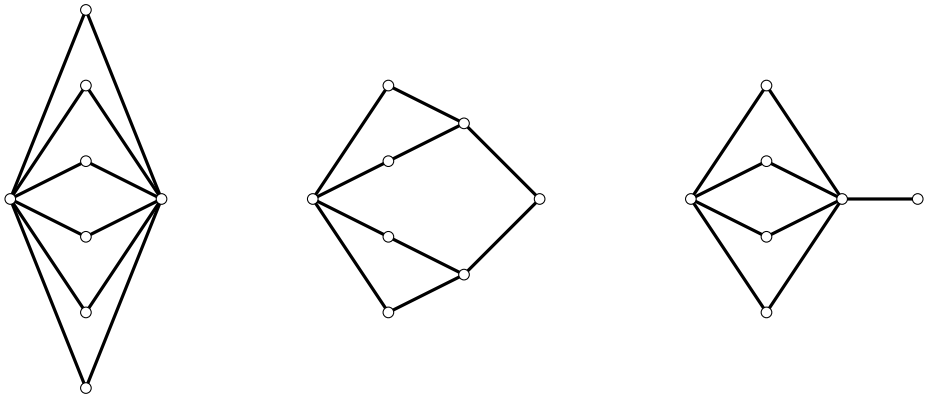
This, non-uniqueness goes beyond the one hidden layers example. What makes the problem tricky for ReLU networks is that the breakpoints (angular points of the piecewise-linear) can be "realized" by different nodes. To visualize this we can look at this example. Let $f : \mathbb{R} \to \mathbb{R}$ be the function in Figure 1. This function can be realized with at least those three networks Φ_1, Φ_2, Φ_3

$$\begin{aligned}
\Phi_1(x) &= \rho(x) - 2\rho(x-1) + \rho(x-2) + \rho(x-4) - 2\rho(x-4) + \rho(x-6) \\
&= \begin{bmatrix} 1 & -2 & 1 & 1 & -2 & 1 \end{bmatrix} \rho \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1 \\ -2 \\ -4 \\ -5 \\ -6 \end{bmatrix} \right) \\
\Phi_2(x) &= \rho\left(\rho(x) - 2\rho(x-1)\right) + \rho\left(\rho(x-4) - 2\rho(x-5)\right) \\
&= \begin{bmatrix} 1 & 1 \end{bmatrix} \rho \left(\begin{bmatrix} 1 & -2 & 0 & 0 \\ 0 & 0 & 1 & -2 \end{bmatrix} \rho \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1 \\ -4 \\ -5 \end{bmatrix} \right) \right) \\
\Phi_3(x) &= \rho\left(\rho(x) - 2\rho(x-1) + 2\rho(x-3) - 2\rho(x-4)\right) \\
&= \rho \left(\begin{bmatrix} 1 & -2 & 2 & -2 \end{bmatrix} \rho \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1 \\ -3 \\ -5 \end{bmatrix} \right) \right)
\end{aligned}$$

The figure 3 shows a schema of those networks.

In this example Φ_1 is the 1-hidden layer realization of f as mentioned above. In an intuitive way we might say that in Φ_1 every node realizes a different breakpoint. Φ_2 generates some breakpoints in deeper nodes, and, finally Φ_3 even generates the breakpoints in 2,4 and 6 together in one node.

This example is to show that different neural networks with the same realization can widely vary in shape. We may guess that it is very hard, or even impossible, to give a simple condition under which we have uniqueness as was done in [1] for other activations



Φ_1 Φ_2 Φ_3

Figure 2: In this schema we indicate with a line all the weights which are not 0. Those are the schemas of Φ_1, Φ_2, Φ_3, the three different ReLU networks that realize f

3 Problem

We have understood why the uniqueness problem for ReLU networks is very hard. What can we do then? A slightly different approach to understand weather neural networks are a reasonable way to parametrize a function space is to count how many different functions we can realize compared to how many neural networks we are using to realize them. We then see weather the number of neural networks is much larger than the number of functions or if they are somehow similar.

Of course we have infinitely many neural networks that can realize infinitely many functions, but this is not very interesting. To make the problem more reasonable we look at a finite subset of neural networks. To do so, we first need to quantize weights: let's define the following quantization

$$\mathbb{Q}_a^b := \left\{2^{-b}N : N \in \mathbb{Z} \text{ with } |N| \leq 2^{a+b+1} - 1\right\}$$

Where a and b are two parameters to tune the quantization precision and the interval of values covered.

The choice of quantizing the weights is quite natural since it is what happens in computer implementations. Furthermore, quantized neural networks, together with their approximation performance, have been extensively studied in [2] and have proven to be a good tool to study neural networks.

To have a finite set of networks we also need the size of the neural network to be finite. For this we need

Definition 4 (Family of networks)**.** *Given two integers a and b we define:*

$$\mathcal{N}_a^b := \left\{\textit{neural networks } \Phi \textit{ which have weights in } \mathbb{Q}_a^b\right\}$$

Furthermore, given two other integers L, W we define:

$$\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W) := \left\{\Phi \in \mathcal{N}_a^b \mid \mathcal{L}(\Phi) \leq L,\ \mathcal{W}(\Phi) \leq W\right\}$$

Now one can easily see that $\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)<\infty$. But how many functions can we realize with networks in $\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)$? To make this question rigorous we define

$$\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W):=\left\{\langle\Phi\rangle^{\mathrm{ReLU}}\mid\Phi\in\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)\right\}$$

Note that the notation $\langle f\rangle^{\mathrm{ReLU}}$ stands for the realization of f with the ReLU activation function.

We are now interested in the scaling behavior of $\log\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ and $\log\#\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ with regard to W and L. If $\log\#\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ scales slower than $\log\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ we will say that there is a degeneration phenomenon.

One way to check weather our strategy is reasonable or not is by looking at the problem for neural network with activation functions as in [1], where we already know that permutation of rows is the only source of non-uniqueness. In such neural networks we have found that there is no degeneration phenomenon for network with $L\geq 4$. This result can be found in appendix D. Since we consider the case where only the permutation of nodes are allowed a well-behaving case, and in this case there is no degeneration: this gives us hope that the degeneration phenomenon is a good tool to study the redundancy of the parametrization.

4 Main result

We managed to find the scaling behavior of $\log\#\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ and $\log\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ in many different settings. All these results are proven in the appendixes

4.1 Fixing the width

The first simple case is when we fix W to be any value equal to or larger than 1, and we look at scaling behavior with regard to L. In this case we have shown that there is no degeneration phenomenon. Formally we have the following theorem

Theorem 1 (Scaling of depth)**.** *for any $W\geq 1$:*

- $\log\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)\in\Theta(L)$
- $\log\#\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W)\in\Theta(L)$

4.2 Fixing the depth

The problem becomes much harder when instead we fix L and look at the scaling behavior with regard to W. In this case we will have a degeneration phenomenon for $L=2,3$. And we will not have it for $L\geq 5$. Whether it degenerates for $L=4$ or not is still an open question. Formally we have the following theorem

Theorem 2 (Scaling of width)**.** *Depending on the depth L we have the following scaling with regard to the width W:*

	$\log \#\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W)$	$\log \#\mathcal{F}_a^b(\mathcal{L} = L, \mathcal{W} = W)$
$L = 2$	$\Theta(W)$	$\Theta(\log W)$
$L = 3$	$\Theta(W^2)$	$\Theta(W \log W)$
$L = 4$	$\Theta(W^2)$	$??$
$L \geq 5$	$\Theta(W^2)$	$\Theta(W^2)$

4.3 Scaling with the connectivity

Finally, we have investigated those neural networks where we don't require the width or the depth to be bounded, but the connectivity.

Definition 5 (Family of networks with bounded connectivity)**.** *Given an integer M we define:*

$$\mathcal{N}_a^b(\mathcal{M} = M) := \left\{\Phi \in \mathcal{N}_a^b \mid \mathcal{M}(\Phi) \leq M\right\}$$
$$\mathcal{F}_a^b(\mathcal{M} = M) := \left\{\langle\Phi\rangle^{ReLU} \mid \Phi \in \mathcal{N}_a^b(\mathcal{M} = M)\right\}$$

Where we remember that the connectivity $\mathcal{M}(\Phi)$ indicates the number of non-zero parameters.

Such, so called "sparse", networks have been the main focus of [2], where they have shown to be able to approximate well a wide variety of functions. We are happy that we could prove that also these networks don't degenerate. Formally:

Theorem 3 (Scaling of connectivity)**.** *For any value a, b:*

- $\log \#\mathcal{F}_a^b(\mathcal{M} = M) \in \Theta(M \log M)$
- $\log \#\mathcal{N}_a^b(\mathcal{M} = M) \in \Theta(M \log M)$

5 Conclusion

The problem of equivalent ReLU networks is hard and far from solved. Assuming that as long as there is no degeneration phenomenon the redundancy problem is not a real issue, we have found a good set of constraints in which the neural networks are a good tool parametrize functions.

Furthermore, our paper gives yet another theoretical evidence on deeper networks performing better than shallow ones ($L \leq 3$ or 4). However, we found very interesting that if $L \geq 5$, scaling the width, no degeneration occurs. This suggests that there might be a radical difference in approximation ability from shallow networks and not-so-shallow ones. An investigation in this direction might also be interesting.

Bibliography

[1] V. Vlačić and H. Bölcskei, "Neural network identifiability for a family of sigmoidal nonlinearities," 2020.

[2] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen, "Optimal approximation with sparsely connected deep neural networks," 2018.

A Fixing the width

Proof of Theorem 1. We divide the theorem in two main steps and a third step for the conclusion:

Step 1: (Scaling of the number of networks) Here we want to show that the log-cardinality of $\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ scales like $\mathcal{O}(L)$.

To show this it suffices to notice that the number of parameter to describe a single linear application of a neural network of width W is no more than W^2+W: W^2 parameters will be for the linear application and W for the bias. From this, simply follows that the number of parameters in a neural network of depth $\leq L$ and width $\leq W$ is no more than $(W^2+W)L$. If all the parameter are picked from $\mathbb{Q}_a^b$ than we have that

$$\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)\leq(\#\mathbb{Q}_a^b)^{(W^2+W)L}.$$

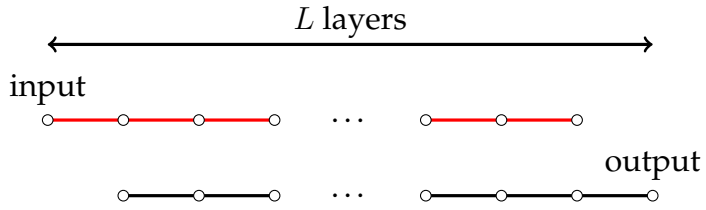
Hence, if we take the logarithm, and we look at the scaling with L we get

$$\log\#\mathcal{N}_a^b(\mathcal{L}=L,\mathcal{W}=W)\leq L\underbrace{(W^2+W)\log\#\mathbb{Q}_a^b}_{\text{doesn't depend on } L}\in\mathcal{O}(L).$$

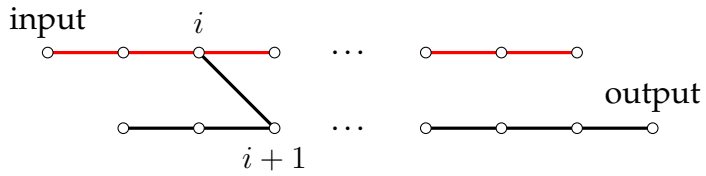
Step 2: (Scaling of the number of realizations) Here we want to show that the log-cardinality of $\log\#\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ scales at least like $\Omega(L)$. To do so we find $\mathcal{S}(L)$, a subset of $\log\#\mathcal{F}_a^b(\mathcal{L}=L,\mathcal{W}=W)$ whose cardinality still scales like $\Omega(L)$.

To make the proof more clear we assume that $W\geq 2$. The proof for $W=1$ is less intuitive and will be presented later. We now define Φ_0 as in the schema below:

— weight = 1
— weight = 2



Of course the $\Phi_0(x) = 0 \ \forall x$. Now let $b \in \{0,1\}^{L-1}$, we then define. We then set at first $\Phi_b = \Phi_0$. Then, for all i such that $b_i = 1$, we add a weight of magnitude 1 between the i-th node in the first row of Φ_b and the $i+1$-th node in the second row as in the figure below:



The final result is that

$$\Phi_b(x) = \left(\sum_{i=1}^{L-1} \mathbb{I}(b_i = 1) 2^{i-1} \right) \text{ReLU}(x).$$

Since $\left(\sum_{i=1}^{L-1} \mathbb{I}(b_i = 1) 2^{i-1} \right)$ is the number which has b as binary representation, then we have that:

$$b_1 \neq b_2 \Rightarrow \Phi_{b_1} \neq \Phi_{b_2}$$

Hence, the cardinality of all possible Φ_b is no less than the cardinality of all possible b, which is 2^{L-1}. From this follows that

$$\log \#\mathcal{F}_a^b(\mathcal{L} = L, \mathcal{W} = W) \geq \log 2^{L+1} = L \log 2 + \log 2 \in \Omega(L).$$

Step 3: (Conclusion) Finally, since the number of all possible realization is smaller than then number of all the neural networks we have intuitively that

$$\Omega(L) \leq \log \#\mathcal{F}_a^b(\mathcal{L} = L, \mathcal{W} = W) \leq \log \#\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W) \leq \mathcal{O}(L).$$

Formally, we have:

- $\log \#\mathcal{F}_a^b(\mathcal{L} = L, \mathcal{W} = W) \in \Theta(L)$
- $\log \#\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W) \in \Theta(L)$

□

As mentioned before, one weakness of this proof is that the Φ_b have width 2. Therefore, our proof only works if $W \geq 2$. We can also show that the result holds $W = 1$, however the construction is not as straightforward, and therefore we decided not to include it in the main proof.

Construction with $W = 1$: The construction uses the same idea of $b \in \{0,1\}^{L-1}$. Now, however, Φ_b only have one row where we set all weights to be equal to 2. The bias of layer i will be -1 if $b_i = 1$, 0 otherwise.

We claim that the network Φ_b has a unique breakpoint $\bar{x}$

$$\bar{x} = \sum_{i=1}^{L-1} \mathbb{I}(b_i = 1) 2^{-i}$$

The proof of this just require a visualization of intermediate layers of Φ_b. What is most relevant is that $\sum_{i=1}^{L-1} \mathbb{I}(b_i = 1)2^{-i}$ is still uniquely determined by b, therefore we can use this new set of Φ_b instead of the one in the proof without any difference.

B Fixing the depth

In this section we tackle the proof of theorem 2. We will decompose the theorem in many prepositions and then look at one problem at the time.

Proof of Theorem 2. The proof of the problem is decomposed as follows:

- In preposition 1 we compute the scaling of the log-cardinality of $\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W)$ on W with all different values of L.
- In preposition 2 we show that $\log \#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) \in \Theta(\log W)$
- In preposition 3 we show that $\log \#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \in \Theta(W \log W)$
- In preposition 4 we show that $\log \#\mathcal{F}_a^b(\mathcal{L} = 5, \mathcal{W} = W) \in \Theta(W^2)$

Of course one can always add an identity layer at the end of a network, hence, if $L > 5$ we will still have that $\mathcal{F}_a^b(\mathcal{L} = 5, \mathcal{W} = W) \in \Theta(W^2)$ □

Preposition 1 (Scaling of $\mathcal{N}$)**.**

$$\log \#\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W) = \begin{cases} W & L = 2 \\ W^2 & L \geq 3 \end{cases}$$

Proof. In the case $L = 2$ Than our neural network can be written as

$$\Phi(x) = \sum_{i=1}^{W} c_i \text{ReLU}(a_i x + b_i).$$

We only have as parameters a_i, b_i, c_i with $i = 1, \ldots, W$. Hence, the total number of parameter is $3W$. We than have that

$$\log \#\mathcal{N}_a^b(\mathcal{L} = 2, \mathcal{W} = W) = \log(\#\mathbb{Q}_a^b)^{3W} = 3W \log \#\mathbb{Q}_a^b \in \Theta(W).$$

If instead $L > 2$, in the intermediate layers, there will be linear applications from W to W. Such applications are parametrized with W^2 parameters. Hence, we have that

$$L \geq 3 \Rightarrow \log \#\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W) = \log(\#\mathbb{Q}_a^b)^{(L-1)W^2} = (L-1)W^2 \log \#\mathbb{Q}_a^b \in \Theta(W^2).$$

□

To prove preposition 2 and 3 we make use of the following lemma

Lemma 1. *Given a vector space X, given $G = \{g_i\}_{i=1}^{\#G}$ finite subset of X, $M \in \mathbb{N}$. For any discretization $\mathbb{Q}_a^b$ we call*

$$\mathcal{F}_a^b := \left\{ \sum_{i=1}^{M} a_i x_i \mid a_i \in \mathbb{Q}_a^b, x_i \in G \right\}.$$

Then we have that

$$\#\mathcal{F}_a^b \leq \min\left\{ (2\#\mathbb{Q}_a^b \#G)^M, (2\#\mathbb{Q}_a^b M)^{\#G} \right\}.$$

Proof. The fact that $\#\mathcal{F}_a^b \leq (2\#\mathbb{Q}_a^b \#G)^M$ can be simply found counting the number of parameters in $\mathcal{F}_a^b$.

To find the other bound we define

$$\mathcal{G}_a^b := \left\{ \sum_{i=1}^{\#G} a_i g_i \mid a_i \in \mathbb{S}_a^b(M), g_i \in G \right\}$$

Where $\mathbb{S}_a^b(M) := \left\{ b a_i \mid b \in 1, \dots, M, a_i \in \mathbb{Q}_a^b \right\}$. One can notice that $\mathcal{F}_a^b \subseteq \mathcal{G}_a^b$.

Furthermore, we know that in $\mathbb{Q}_a^b$ values are equispaced and $0 \in \mathbb{Q}_a^b$, hence we have that $\#\mathbb{S}_a^b(M) = M\#\mathbb{Q}_a^b$. Counting parameters we can conclude that

$$\#\mathcal{F}_a^b \leq \#\mathcal{G}_a^b \leq (2\#\mathbb{Q}_a^b M)^{\#G}.$$

□

Preposition 2 (Scaling with $L = 2$)**.**

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) \in \Theta(\log W)$$

Proof. Since we need to find a Θ-scaling, we need both the lower and the upper bound.

Step 1: (upper bound) To find a lower bound we start by considering

$$G_1 := \left\{ x \in \mathbb{R} \to \text{ReLU}(dx + e) \mid d, e \in \mathbb{Q}_a^b \right\}.$$

Since a and b are the only parameters, it's easy to notice that $\#G_1 = (\#\mathbb{Q}_a^b)^2$. Now one can rewrite

$$\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) = \left\{ \sum_{i=1}^{W+1} a_i f_i \mid a_i \in \mathbb{Q}_a^b, f_i \in G_1 \right\}.$$

Now, thanks to lemma 1, having $M = W + 1$ and $G = G_1$, we get that

$$\#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) \leq (2\#\mathbb{Q}_a^b(W+1))^{\#G_1} = (2\#\mathbb{Q}_a^b(W+1))^{(\#\mathbb{Q}_a^b)^2}.$$

Finally if we look at the log-cardinality we can get the desired bound:

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) = \underbrace{(\#\mathbb{Q}_a^b)^2 \log\left(2\#\mathbb{Q}_a^b\right)}_{\text{doesn't depend on } W} \log(W+1) \in \mathcal{O}(\log W)$$

Step 2: (lower bound) to find a lower bound we just need to notice that the function below can be realized with a network in $\mathcal{N}_a^b(\mathcal{L} = 2, \mathcal{W} = W)$:

$$\left\{ \sum_{i=1}^{K} a\text{ReLU}(bx) = Ka\text{ReLU}(bx) \mid K \in 1, \dots, M, a, b \in \mathbb{Q}_a^b \right\} \subseteq \mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W)$$

Of course, even in this subset we can have W different functions ($K = 1, \dots, W$). Therefore $\#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) \geq W$. Therefore, looking at log-cardinalities

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) \in \Omega(\log W).$$

Having found both a lower and an upper bound we have proven that

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W) \in \Theta(\log W).$$

Note: before concluding the proof we notice that no properties of the ReLU function where exploited, hence, the proof also holds for any other activation function. □

Preposition 3 (Scaling with $L = 3$)**.**

$$log\#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \in \Theta(W \log W)$$

Proof. Since we need to find a Θ-scaling, we need both the lower and the upper bound.

Step 1: (upper bound) To find this bound we start by considering

$$G_2 := \left\{ x \in \mathbb{R} \to \text{ReLU}(f(x)) \mid f \in \mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \right\}.$$

The cardinality of G_2 is of course the same of $\mathcal{F}_a^b(\mathcal{L} = 2, \mathcal{W} = W)$, using the bound we have found in the proof of preposition 2, we can bound $\#G_2$:

$$\#G_2 \leq (2\#\mathbb{Q}_a^b(W+1))^{(\#\mathbb{Q}_a^b)^2}$$

Now one can write

$$\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) = \left\{ \sum_{i=1}^{W+1} a_i, f_i \mid a_i \in \mathbb{Q}_a^b, f_i \in G_2 \right\}.$$

We can finally use lemma 1 with $M = W + 1$ and $G = G_2$ to get

$$\#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \leq (2\#\mathbb{Q}_a^b \#G_2)^{W+1} = (2\#\mathbb{Q}_a^b)^{W+1} (2\#\mathbb{Q}_a^b(W+1))^{(\#\mathbb{Q}_a^b)^2(W+1)}.$$

Looking at the log-cardinality we get

$$\begin{aligned} \log \#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) &\leq (W+1)\log(2\#\mathbb{Q}_a^b) + (W+1)(\#\mathbb{Q}_a^b)^2(\log 2 + \log \#\mathbb{Q}_a^b + \log(W+1)) \\ &\in \mathcal{O}(W \log W). \end{aligned}$$

The second row follows from the fact that $W \log W$ is the dominating term.

Step 2: (lower bound) To find a lower bound we start by fixing a $d > 0 \in \mathbb{Q}_a^b$. Now we can look at

$$\mathcal{H}_a^b = \left\{ \sum_{i=1}^{M} \text{ReLU}(a_i dx + b_i d) \mid a_i, b_i \in 1, \dots, M \right\}.$$

I would like to notice that $\mathcal{H}_a^b \subseteq \mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = 3W)$. To make this more obvious we can rewrite elements of $\mathcal{H}_a^b$ as follows:

$$\sum_{i=1}^{M} \text{ReLU}(a_i dx + b_i d) = \sum_{i=1}^{W} \text{ReLU}\left(\sum_{j=1}^{a_i} (\text{ReLU}(x) - \text{ReLU}(-x)) + \sum_{j=1}^{b_i} 1 \right)$$

It's important to notice that the vector of a_i and b_i don't uniquely determinate a function in $\mathcal{H}_a^b$. Yet, if we know that two functions have different breakpoints, then they must be different. The question is now, in how many ways can we pick W breakpoints? The breakpoints will be in $\left\{ -\frac{a}{b} \mid a, b \in 1, \dots, M \right\}$. We will call $C(W) := \# \left\{ -\frac{a}{b} \mid a, b \in 1, \dots, M \right\}$.

We can draw M different breakpoint, therefore we can do this in $\binom{C(W)}{W}$ different ways.

Hence, looking at log-cardinalities, we have that

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \geq \log \binom{C(W)}{W} \geq \log \left(\frac{C(W)}{W} \right)^W \in \Omega \left(W \log \frac{C(W)}{W} \right).$$

To conclude the proof we use a result of number theory cite some number theory result that tells us that $C(W) \sim W^2$. Hence, $\log \#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \in \Omega\left(W \log W\right)$.

Since we have both an upper and a lower bound we can conclude that

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 3, \mathcal{W} = W) \in \Theta\left(W \log W\right)$$

Note: before concluding the proof we notice that no properties of the ReLU function where exploited, hence, the proof also holds for any other activation function. □

Preposition 4 (Scaling with $L = 5$)**.**

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 5, \mathcal{W} = W) \in \Theta(W^2)$$

Proof. For the proof of this preposition it is sufficient to prove the upper bound. The lower bound follows from the fact that the number of realizations must be lower than the number of networks:

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 5, \mathcal{W} = W) \leq \log \#\mathcal{N}_a^b(\mathcal{L} = 5, \mathcal{W} = W) \in \mathcal{O}(W^2)$$

To find an upper bound the proof is more complicated. We start by taking $M \in \mathbb{N}$. To follow the proof, it helps to think that there exists a c such that $M = cW$.

Step 1: (basis functions) We start by defining M functions:

$$\psi_k(x) = \begin{cases} 0 & x \leq 2k \\ (x-2k)M & 2k \leq x \leq 2k+1 \\ (2k+2-x)M & 2k+1 < x < 2k+2 \\ 0 & x \geq 2k+2 \end{cases} \qquad k = 1, \dots, M$$

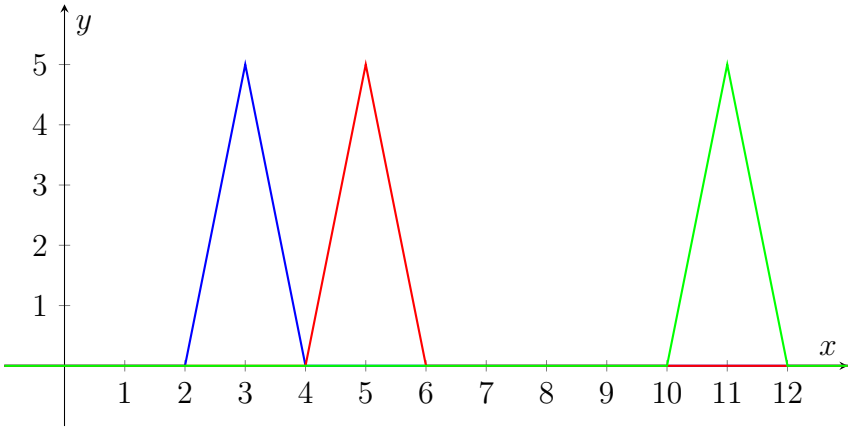


Figure 3: Example with $M = 5$ of 3 different ψ_k functions for $k = 1, 2, 5$

Now we define another set of M^2 functions:

$$\varphi_k^h(x) = \text{ReLU}(\psi_k(x) - h + 1) \qquad k, h \in 1, \dots, M$$

I would like to notice that all the φ_k^h have breakpoints in different points. This might not be trivial at first sight, but to get a good intuition one might look at figure 4. Since by linearly combining continuous piece-wise function one cannot add breakpoints, we conclude that all φ_k^h are linearly independent.

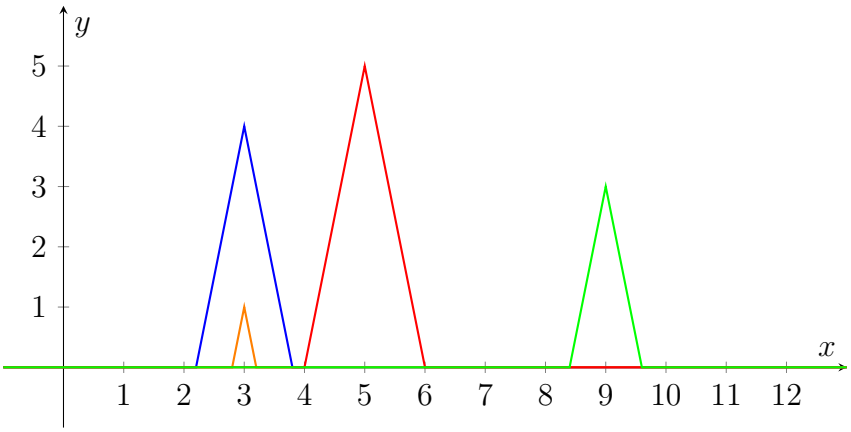


Figure 4: Example with $M = 5$ of 4 different φ_k^h functions

Step 2: (realization of the basis) We now want to realize all the φ_k^h with a neural network (where the width scales linearly with M). To do so, we start by decomposing ψ_k in a_k, b_k and c_k:

$$\begin{aligned}\psi_k(x) &= \text{ReLU}(M(x-2k)) - \text{ReLU}(2M(x-(2M+1))) + \text{ReLU}(M(x-(2k+2))) \\ &= \underbrace{\text{ReLU}(Mx-2Mk)}_{:=a_k(x)} - \underbrace{\text{ReLU}(2Mx-4Mk-2M)}_{:=b_k(x)} + \underbrace{\text{ReLU}(Mx-2Mk-2M)}_{:=c_k(x)}\end{aligned}$$

Then we can write the φ_k^h functions as:

$$\varphi_k^h(x) = \text{ReLU}(a_k(x) - b_k(x) + c_k(x) - h + 1)$$

We now build a network $\Upsilon_M(x)$ that, with a width linear in M and a depth of 3, realizes

$$\Upsilon_M(x) = A_3 \circ \text{ReLU} \circ A_2 \circ \text{ReLU} \circ A_1 = \begin{bmatrix} a_1(x) \\ b_1(x) \\ c_1(x) \\ \vdots \\ a_M(x) \\ b_M(x) \\ c_M(x) \\ 1 \\ 2 \\ \vdots \\ M \end{bmatrix}.$$

Now, we show how we can choose affine transformations A_1, A_2, A_3 in such a way that they realize Υ_M. To make this more simple we introduce this notation:

- $\mathbf{0}_{m,n}$ indicates an $m \times n$ matrix where all the entries are 0
- $\mathbf{1}_{m,n}$ indicates an $m \times n$ matrix where all the entries are 1.
- $\mathbf{L}_m$ indicates a square lower triangular matrix of 1s of size m. Formally

$$[\mathbf{L}_m]_{i,j} = \begin{cases} 1 & i \leq j \\ 0 & i > j \end{cases}$$

$$A_1 : \mathbb{R} \to \mathbb{R}^{5M}$$

$$A_1(x) = \begin{bmatrix} \mathbf{1}_{M,1} \\ \mathbf{0}_{M,M} \\ \mathbf{0}_{M,M} \\ \mathbf{0}_{M,M} \\ \mathbf{0}_{M,M} \end{bmatrix} x + \begin{bmatrix} \mathbf{0}_{M,1} \\ \mathbf{1}_{M,1} \\ \mathbf{1}_{M,1} \\ \mathbf{1}_{M,1} \\ \mathbf{1}_{M,1} \end{bmatrix} = \begin{bmatrix} x \\ x \\ \vdots \\ x \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$A_2 : \mathbb{R}^{5M} \to \mathbb{R}^{3M+2}$$

$$A_2(x) = \begin{bmatrix} \mathbf{1}_{2,M} & \mathbf{0}_{2,M} & \mathbf{0}_{2,M} & \mathbf{0}_{2,M} & \mathbf{0}_{2,M} \\ \mathbf{0}_{M,M} & \mathbf{1}_{M,M} & \mathbf{1}_{M,M} & \mathbf{0}_{M,M} & \mathbf{0}_{M,M} \\ \mathbf{0}_{M,M} & \mathbf{1}_{M,M} & \mathbf{1}_{M,M} & \mathbf{1}_{M,M} & \mathbf{1}_{M,M} \\ \mathbf{0}_{M,M} & \mathbf{L}_{M,M} & \mathbf{0}_{M,M} & \mathbf{0}_{M,M} & \mathbf{0}_{M,M} \end{bmatrix} x + \mathbf{0}_{3M+2,1}$$

Now, if we combine $A_2 \circ \text{ReLU} \circ A_1$ we get:

$$A_2 \circ \text{ReLU} \circ A_1(x) = \begin{bmatrix} M\text{ReLU}(x) \\ M\text{ReLU}(x) \\ 2M \\ \vdots \\ 2M \\ 4M \\ \vdots \\ 4M \\ 1 \\ 2 \\ \vdots \\ M \end{bmatrix}$$

If we then look at Υ_M we notice that all the ingredients for it are ready in $A_2 \circ \text{ReLU} \circ A_1$. We need to remember that considering $\text{ReLU}(x)$ and x is equivalent since the a_k, b_k and c_k functions take the ReLU of the input. One can now understand that we can build an A_3, using only $-1, 0, 1$ as entries, such that $A_3 \circ \text{ReLU} \circ A_2 \circ \text{ReLU} \circ A_1 = \Upsilon_M(x)$.

Finally, we notice that we can simply compute $\psi_k - h + 1, \ \forall k, h \in 1, \ldots, M$ as linear combination of the $\Upsilon_M(x)$. Applying another layer, and therefore a ReLU nonlinearity, one could compute any φ_k^h. However, we need to do so in a smart way, otherwise, if we realize only one φ_k^h per node in the next layer we don't have an M^2 scaling in the log-cardinality. The intuitive idea of the next step is to realize together all the φ_k^h that have the same h.

Step 3: (the 2^{M^2} scaling) We start by defining $B \in \{0,1\}^{M \times M}$. We then define

$$\Phi_B(x) := \sum_{h=1}^{M} \sum_{k=1}^{M} B_{h,k} \varphi_k^h(x).$$

Since all the $\varphi_k^h(x)$ are linearly independent, we have that

$$B \neq \tilde{B} \Leftrightarrow \Phi_B \neq \Phi_{\tilde{B}}.$$

Since B has M^2 entries, and all the entries can be either 0 or 1, we can draw B in 2^{M^2} different ways.

Step 4: (realization of Φ_B) Here we start from $\Upsilon_M(x)$, we then concatenate other two layers, and then we can get any $\Phi_B(x)$. Of course the layers we add must depend on B. The final (5-th) layer is trivial and just sums up all the outputs of the previous one, the key part is in the second-to-last (4-th) layer.

First, we notice that ReLU $\circ\, \Upsilon_M$ has no effect since all the entries of Υ_M are non-negative. Then we rewrite Φ_B

$$\Phi_B(x) = \sum_{h=1}^{M} \underbrace{\sum_{k=1}^{M} B_{h,k}\varphi_k^h(x)}_{:=\eta_B^h(x)} = \sum_{h=1}^{M} \eta_B^h(x).$$

We will build the 4-th layer in such a way that the h-th entry will be $\eta_B^h(x)$. A visualization of this is in figure 5. To have the h-th node computing $\eta_B^h(x)$, we need to rewrite $\eta_B^h(x)$ in the following way:

$$\begin{aligned}
\eta_B^h(x) &= \sum_{k=1}^{M} B_{h,k}\varphi_k^h(x) \\
&= \sum_{k=1}^{M} B_{h,k}\text{ReLU}(\psi_k(x) - h + 1) \\
&\overset{(i)}{=} \text{ReLU}\left(\sum_{k=1}^{M}(B_{h,k}\psi_k(x)) - h + 1\right) \\
&= \text{ReLU}\left(\sum_{k=1}^{M}(B_{h,k}a_k(x) + (-B_{h,k})b_k(x) + B_{h,k}c_k(x))) + (-1)h + 1\right)
\end{aligned}$$

To understand why (i) holds, we check that it's true for $x \in [2k, 2k+2]\, \forall k = 1, \ldots, M$. If $x \in [2k, 2k+2]$, then $\psi_{\tilde{k}}(x) = 0,\ \tilde{k} \neq k$. Now, foreach k, we new show that the

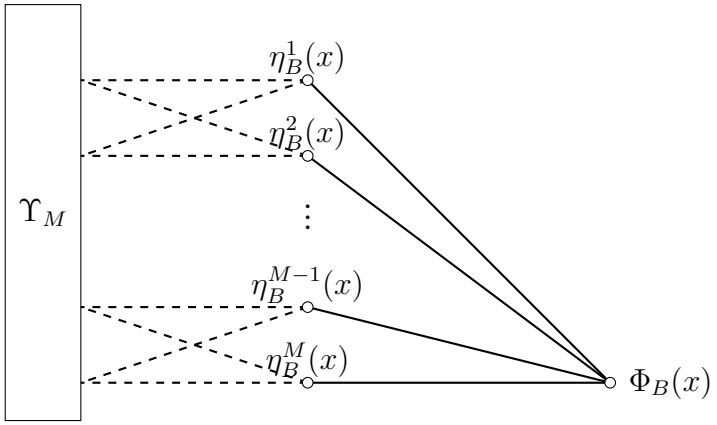


Figure 5: Schema of the 2 last layer of $\Phi_B(x)$. The dashed lines from Υ_M to the 4-th layer represent a general affine transformation

equality (i) holds in $[2k, 2k+2]$

$$\begin{aligned}
\text{ReLU}\left(\sum_{\tilde{k}=1}^{M}(B_{h,\tilde{k}}\psi_{\tilde{k}}(x)) - h + 1\right) &= \text{ReLU}\left(\underbrace{\sum_{\tilde{k}\neq k}(B_{h,\tilde{k}}\psi_{\tilde{k}}(x))}_{=0} + B_{h,k}\psi_k(x) - h + 1\right) \\
&= \text{ReLU}\left(B_{h,k}\psi_k(x) - h + 1\right) \\
&= \text{ReLU}\left(B_{h,k}\psi_k(x) - h + 1\right) + \underbrace{\sum_{\tilde{k}\neq k}\text{ReLU}\left(B_{h,\tilde{k}}\psi_{\tilde{k}}(x) - h + 1\right)}_{=0} \\
&= \sum_{\tilde{k}=1}^{M} B_{h,\tilde{k}}\text{ReLU}(\psi_{\tilde{k}}(x) - h + 1)
\end{aligned}$$

Now, it's trivial to see that the equality also holds in $(infty, 2) \cup (2k+2, \infty)$, since both the members are 0. Hence, we conclude that (i) is true for any value of x.

Once we have rewritten $\eta_B^h(x)$ as above, it's easy to see that is simply a ReLU function applied to an affine transformation of Υ_M using only weights in $\{-1, 0, 1\}$.

Now that h-th entry of the 4-th layer is $\eta_B^h(x)$. It's then trivial to see that the whole network, which just sums all the note of the 4-th layer, will be $\Phi_B(x)$.

Step 5: (conclusion) We have now managed to build a network for any Φ_B with weights in $\{-1, 0, 1\} = \mathbb{Q}_a^b$, depth 5, and width $4M$. Therefore, we can conclude that

$$\#\mathcal{F}_a^b(\mathcal{L} = 5, \mathcal{W} = 5M) \geq \#\{\Phi_B \mid B \in \{0,1\}^{M\times M}\} = 2^{M^2}.$$

Hence, looking at log-cardinalities and setting $W = 5M$,

$$\log \#\mathcal{F}_a^b(\mathcal{L} = 5, \mathcal{W} = W) \geq \left(\frac{W}{5}\right)^2 \log 2 \in \Omega(W^2).$$

Since we have found the lower bound before, this concludes the proof. □

C Scaling with the connectivity

Before starting with the proof of theorem 3 we would like to make two remarks:

- The theorem is proven for $b = 0$ and $a \geq 2$. It's trivial to generalize for $b > 0$ and I think it's not so interesting to generalize for $a = 1$ (but I still think that the result holds anyway)
- When we refer to $\mathcal{N}_a^b(\mathcal{M} = M)$ we require that the nodes are somehow all linked together. (This is both obvious and reasonable)

Proof of Theorem 3. The proof is done assuming $a \geq 2$ and $b = 0$. ($b = 0$ is without loss of generality since b has only to do with scaling).

- Thanks to Preposition 5 we have that $\log \#\mathcal{F}_a^0(\mathcal{M} = M) \in \Omega(M \log M)$
- Thanks to Preposition 6 we have that $\log \#\mathcal{N}_a^0(\mathcal{M} = M) \in \mathcal{O}(M \log M + M \log \#\mathbb{Q}_a^0)$. Since in this theorem we are interesting at the scaling behavior with M rather than with $\#\mathbb{Q}_a^0$ then we can say that $\log \#\mathcal{N}_a^0(\mathcal{M} = M) \in \mathcal{O}(M \log M)$ (the second term is only $\mathcal{O}(M)$)

Since for each network in $\mathcal{N}_a^0(\mathcal{M} = M)$ there is a realization in $\mathcal{F}_a^0(\mathcal{M} = M)$, but, on the other hand, multiple nets of $\mathcal{N}_0^b(\mathcal{M} = M)$ might have the same realization we have

$$\#\mathcal{F}_a^0(\mathcal{M} = M) \leq \#\mathcal{N}_a^0(\mathcal{M} = M).$$

From this it is easy to show that

$$\log \#\mathcal{N}_a^0(\mathcal{M} = M) \in \begin{cases} \mathcal{O}(M \log M) \\ \Omega(M \log M) \end{cases} \Rightarrow \log \#\mathcal{N}_a^0(\mathcal{M} = M) \in \Theta(M \log M).$$

In the same way one can show that $\log \#\mathcal{F}_a^0(\mathcal{M} = M) \in \Theta(M \log M)$. □

Preposition 5. *For $a \geq 2$*

$$\log \#\mathcal{F}_a^0(\mathcal{M} = M) \in \Omega(M \log M)$$

Proof. The proof is done in 3 steps. To follow the proof more easily keep in mind that $M = c\sqrt{N}$ for some constant c.

Step 1: (Define Q_N and look at its subsets)

$$Q_N := \left\{ z_{ijk} = \frac{5^i}{2^j 3^k} \mid i, j, k = 0, \ldots, N-1 \right\}$$

And then we define the set of all interesting subsets

$$\mathcal{P}_N := \left\{ P \subseteq Q_N \mid \#P \leq N^2 \right\}.$$

Note: this is useful since we will find a neural network Φ_P with bounded connectivity for each $P \in \mathcal{P}_N$

It's easy to see that $\#Q_N = N^3$ since $2, 3, 5$ are prime. Now we compute the log-cardinality of $\mathcal{P}_N$:

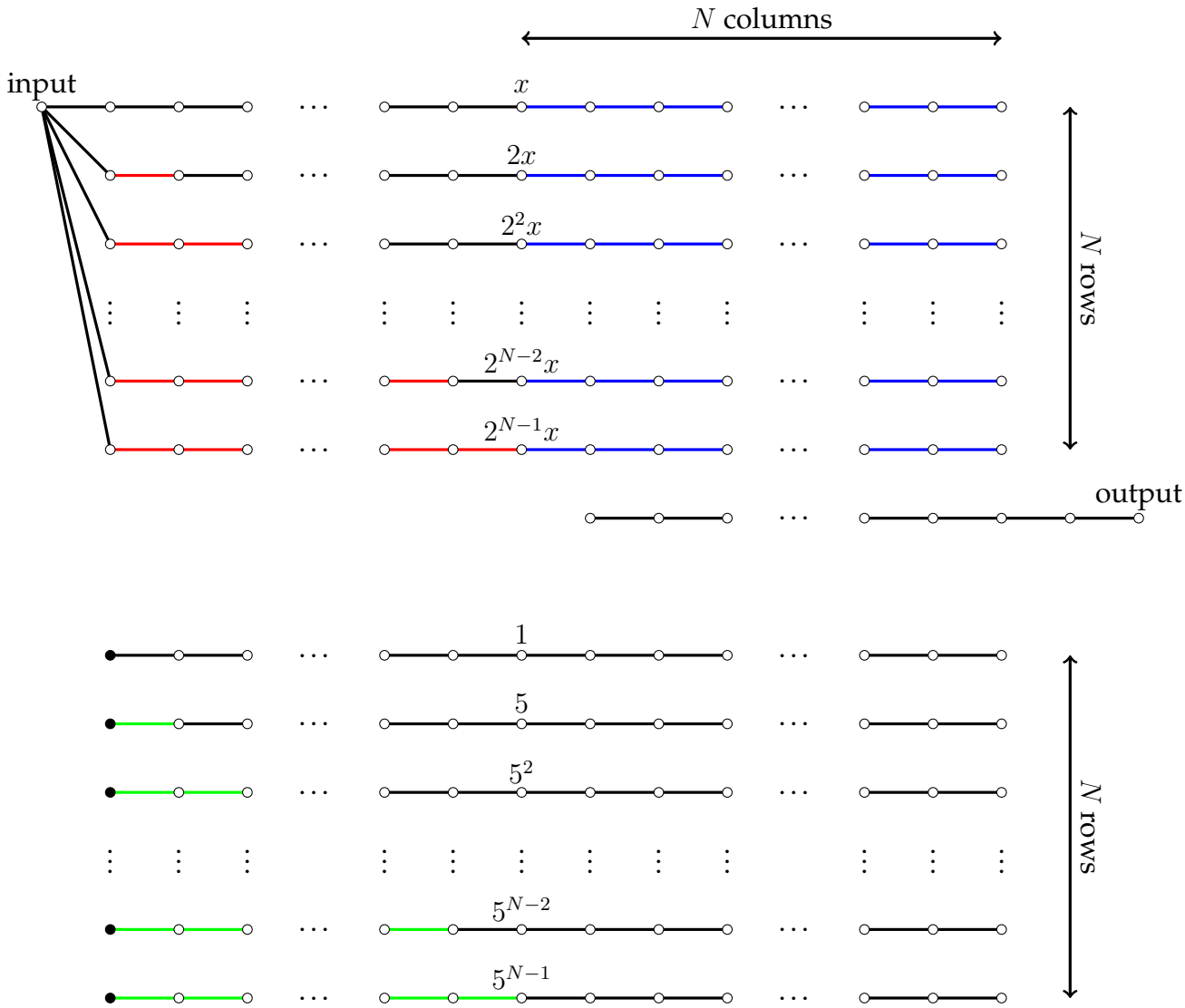
$$\log \#\mathcal{P}_N = \log \binom{\#Q_N}{N^2} \overset{(1)}{\geq} \log \left(\frac{N^3}{N^2}\right)^{N^2} = N^2 \log N$$

Where (1) follows from a well known bound for the binomial coefficient: $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$

Step 2: (find a distinct neural network Φ_P for each $P \in \mathcal{P}_N$ with connectivity in the order of $\mathcal{O}(N^2)$)

We start by building a skeleton neural network Φ like in the next figure:

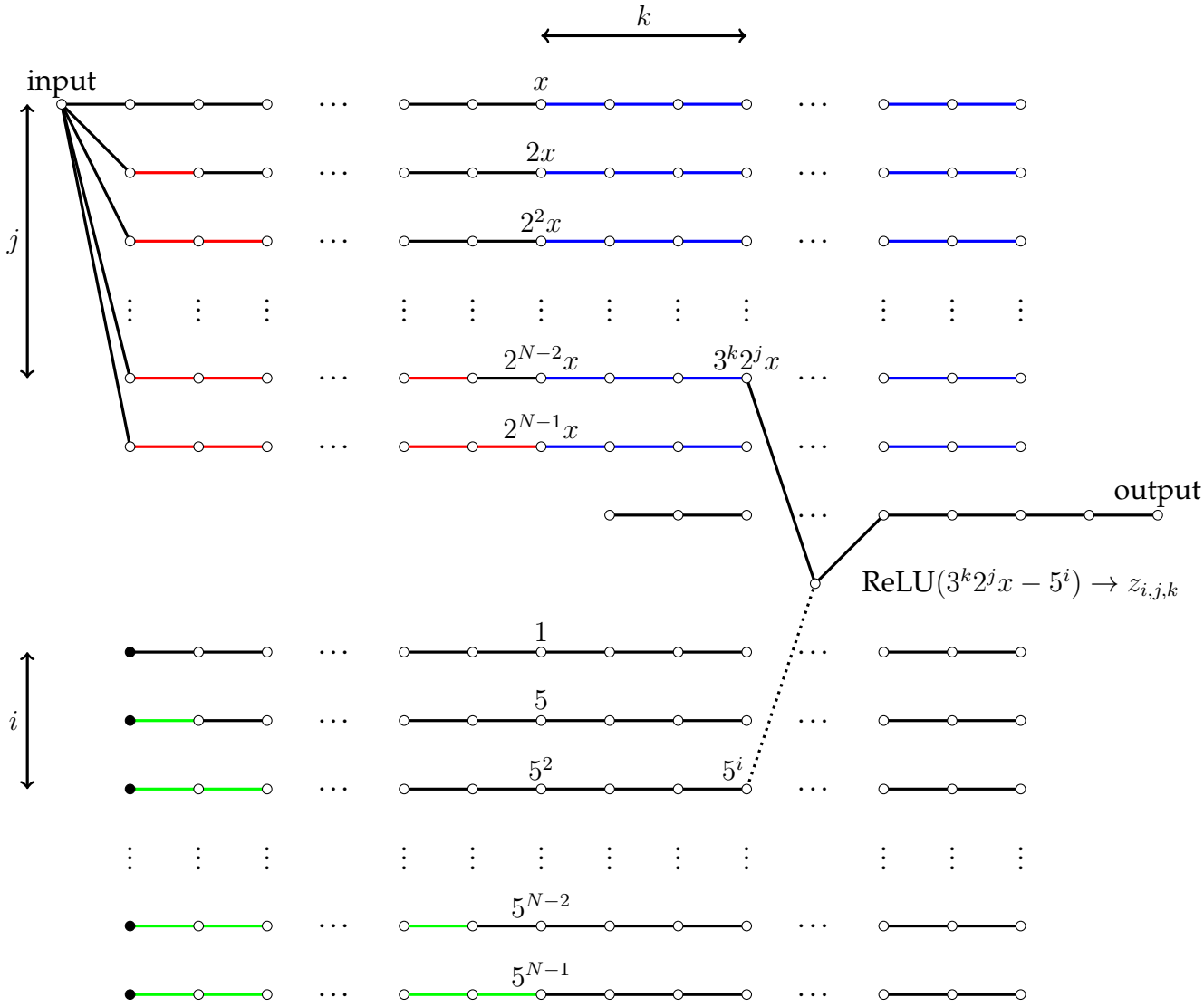
— weight = 1
— weight = 2
— weight = 3
— weight = 5
∘ no bias in this neuron
• bias = 1



Note that this network has no output, $(\Phi = 0)$. We will fix this later on.

Furthermore, note that the connectivity of this network is $\mathcal{M}(\Phi) = 3N + 4N^2$.

Now we can show that by adding 3 weight to Φ we can add a breakpoint in Φ chosen from Q_N. Let $z_{i,j,k}$ be the breakpoint: (dashed line correspond to weight $= -1$)



Let $P = \left\{z^{(1)}, \dots z^{(m)}\right\}, m \leq N^2$. To realize Φ_P we start from Φ and then add breakpoints $z^{(1)}, z^{(2)}, \dots, z^{(m)}$ one at the time.

Now we can notice 2 properties:

1. $\mathcal{M}(\Phi_P) \leq CN^2$ for $C \in \mathbb{N}$. This is true since each new breakpoint only uses 3 weights and the original Φ has around $(2N)^2$ weights. (I think therefore that $C = 8$, but that is not the point)

2. Since Φ_P has breakpoints exactly in P, it's easy to understand that $P \neq P' \Leftrightarrow \Phi_P \neq \Phi_{P'}$. Hence, $\#\left\{\Phi_P \mid P \in \mathcal{P}_N\right\} = \#\mathcal{P}_N$.

Step 3: (Conclusion)

By noticing that $\{\Phi_P \mid P \in \mathcal{P}_N\} \subseteq \mathcal{F}_a^0(\mathcal{M} = CN^2)$ it's easy to see that:

$$\#\mathcal{F}_a^b(\mathcal{M} = CN^2) \geq \#\{\Phi_P \mid P \in \mathcal{P}_N\} = \#\mathcal{P}_N \geq N^2 \log N$$

So if we chose $M = CN^2 \Rightarrow N^2 = \frac{M}{C}$ we have shown that:

$$\#\mathcal{F}_a^b(\mathcal{M} = M) \geq \frac{M}{C} \log \sqrt{\frac{M}{C}} = \frac{M}{2C}\left(\log M + \log \frac{1}{C}\right) \in \Omega\left(M \log M\right)$$

□

Preposition 6.

$$\log \#\mathcal{N}_a^0(\mathcal{M} = M) \in \mathcal{O}(M \log M + M \log \#\mathbb{Q}_a^0)$$

Note: the proof of this theorem is not original since the result is already contained in the bit encoding of a neural network proposed in Preposition 2.34 of [**?**]. However, since the usage of the result is different, I think that rephrasing it here makes it more clear.

Proof. The proof is done by looking at how many different networks we can choose from $\mathcal{N}_a^0(\mathcal{M} = M)$ (in particular we look for an upper bound).

Step 1: (Node distribution) We know the length of a neural network $\Phi \in \mathcal{N}_a^0(\mathcal{M} = M)$ must be less than or equal to M. So let $N_i, i = 1, \ldots, M$ denote the number of nodes in the i-th layer. (we will choose $N_i = 0$ for $i > L$ if Φ has a length L).

It's very easy to see that $0 \leq N_i \leq M \ \forall i = 1, \ldots, M$. Hence:

$$(N_1, \ldots, N_M), N_i \in \{0, \ldots, M\} \text{ can be drawn in } (M+1)^M \text{ different ways}$$

(This bound can be made tighter (A^M) since we also know that $\sum_{i=1}^{M} N_i \leq M$, but it doesn't change the result of the proof since Step 2 is the dominant term)

Step 2: (Weights positioning) Once we fix the positioning of the Nodes $(N_1, \ldots, N_M)$ we want to look at how many ways we can choose which weights are non-zero. We will say that each weight links any couple of nodes, and we will denote bias as weights from one node to itself. (We are considering many configurations which are not legal for $\Phi \in \mathcal{N}_a^0(\mathcal{M} = M)$, but this is not a problem since we are looking for an upper bound).

Since each weight can link any couple of nodes and the total number of nodes is $\leq M$ we have that there are M^2 ways we can draw a weight positioning. Now we need to draw M of those weights:

$$\text{all possible weights positioning can be drawn in } \binom{M^2}{M} \text{ different ways}$$

Now, we use a well known bound for the binomial coefficient,

$$\binom{M^2}{M} \leq \left(\frac{eM^2}{M}\right)^M = e^M M^M.$$

Step 3: (Weights values) Now we need to look at how many values can those weights actually have. Since each weight can be drawn in $\#\mathbb{Q}_a^0$ different ways, in total, all weights can be drawn in $(\#\mathbb{Q}_a^0)^M$ different ways.

Step 4: (Conclusion) By putting all together, $\Phi \in \mathcal{N}_a^0(\mathcal{M} = M)$ can be drawn in less than

$$(M+1)^M e^M M^M (\#\mathbb{Q}_a^0)^M \text{ different ways.}$$

Hence we can bound the log-cardinality, we get

$$\log \#\mathcal{N}_a^0(\mathcal{M} = M) \leq M\log(M+1)+M+M\log M+M\log \#\mathbb{Q}_a^0 \in \mathcal{O}(M\log M+M\log \#\mathbb{Q}_a^0).$$

□

D Non ReLU degeneration proof

In this appendix we want to show that for any activation function ρ for which theorem 1 in [1], then there is no degeneration phenomenon. To do so we start by reintroducing some notation from paper and restate the theorem that we are interested in.

Definition 6 (No-clone set)**.** *We define $\tilde{\mathcal{N}}_a^b(\mathcal{L} = L, \mathcal{W} = W)$ to be the set of all network in $\mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W)$ that respect these two additional conditions*

- *For any layer ℓ, the matrix W_ℓ does not have a zero column or a zero row*
- *For any layer ℓ, there cannot be two rows i, j such that:*

$$(W_{i,1}, \dots, W_{i,D_{\ell-1}}, \theta_i) = (W_{j,1}, \dots, W_{j,D_{\ell-1}}, \theta_j)$$

We now can state the theorem that interests us

Theorem 4. *[Uniqueness theorem from [1]] Let ρ be a piece-wise C^1 function such that:*

$$\rho' \in \left\{ f \in L^1(\mathbb{R}) \mid \|f\|_{BV(\mathbb{E})} := \sup_{\substack{\varphi \in C_c^1(\mathbb{R}) \\ \|\varphi\|_{L^\infty(\mathbb{R})}}} \int_{\mathbb{R}} f(x)\varphi'(x)dx \leq \infty \right\}$$

Let $\epsilon > 0$. Then there exists a function $\sigma : \mathcal{D} \to \mathbb{C}, \mathcal{D} \supset \mathbb{R}, \sigma(\mathbb{R}) \subset \mathbb{R}$ such that all networks in $\tilde{\mathcal{N}}_a^b(\mathcal{L} = L, \mathcal{W} = W)$, together with σ activation function, excluding permutation of nodes, realize a unique function.

We now show that there is no degeneration phenomenon for $L \geq 4$, but this is enough since, from preposition 2 and preposition 3, we already know that there is a degeneration for $L = 2, 3$ for any activation function.

Definition 7 (Non ReLU realization family)**.**

$$\mathcal{F}_a^b(\mathcal{A} = \rho, \mathcal{L} = L, \mathcal{W} = W) := \left\{ \langle \Phi \rangle^\rho \mid \Phi \in \mathcal{N}_a^b(\mathcal{L} = L, \mathcal{W} = W) \right\}$$

Theorem 5 (Non ReLU)**.** *Let ρ be an activation function for which theorem 4 holds. Then, for any $L \geq 4$, we have that*

$$\log \#\mathcal{F}_a^b(\mathcal{A} = \rho, \mathcal{L} = L, \mathcal{W} = W) \in \Theta(W^2)$$

Proof. To prove this theorem we build a family of networks which are inside $\mathcal{F}_a^b(\mathcal{A} = \rho, \mathcal{L} = L, \mathcal{W} = W)$ and for which we can prove that they are all unique using theorem 4.

Step 1: (b vector of vectors) We start by defining

$$B := \left\{ \{b_1, \dots, b_W\} \mid b_i \in \{0,1\}^W, b_i \neq b_j \quad \forall i, j = 1, \dots, W \right\}$$

b_i can be draw in 2^W different ways, then the number of elements in B is

$$\#B = \binom{2^W}{W}$$

Since we need to count in how many ways we can draw W different b vectors from $\{0,1\}^W$. Now we can use a bound for the binomial coefficient to show

$$\log \#B = \log \binom{2^W}{W} \geq \log \left(\frac{2^W}{W} \right)^W = W(\log 2^W - \log W) = W^2 \log 2 - W \log W \in \Omega(W^2).$$

Step 2: (building the network) Without loss of generality we assume that $a = 1, b = 0$, hence we can choose weights in $\{-1, 0, 1\}$. We build a depth-4 network for any $\{b_1, \dots, b_W\} \in B$. We write $\Phi_{b_1,\dots,b_W} = A_4 \circ \rho \circ A_3 \circ \rho \circ A_2 \circ \rho \circ A_1$. A_3 will be the only one that depends on $\{b_1, \dots, b_W\}$.

$$\begin{aligned}
&A_1 : \mathbb{R} \to \mathbb{R}^W \\
&A_1(x) = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\
&A_2 : \mathbb{R}^W \to \mathbb{R}^W \\
&A_2(x) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \cdots & \cdots & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&A_3 : \mathbb{R}^W \to \mathbb{R}^W \\
&A_3(x) = \begin{bmatrix} b_1^T \\ b_2^T \\ \vdots \\ b_M^T \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&A_4 : \mathbb{R}^W \to \mathbb{R} \\
&A_4(x) = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} x + 0
\end{aligned}$$

Now, we can't say that all of those networks have unique realizations since A_1 has repeated rows, hence $\Phi_{b_1,\dots,b_W}$ is not in $\tilde{\mathcal{N}}(\mathcal{L}=4,\mathcal{W}=W)$. But we can modify A_1 and A_2 in such a way that $\Phi_{b_1,\dots,b_W}$ doesn't change, but we have unique rows.

$$\begin{aligned}
&\tilde{A}_1 : \mathbb{R} \to \mathbb{R} \\
&\tilde{A}_1(x) = x + 0 \\
&\tilde{A}_2 : \mathbb{R} \to \mathbb{R}^W \\
&\tilde{A}_2(x) = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ M \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\end{aligned}$$

We here notice that

$$\begin{aligned}
A_2 \circ \rho \circ A_1(x) &= \begin{bmatrix} \rho(x) \\ \rho(x) + \rho(x) \\ \vdots \\ \rho(x) + \cdots + \rho(x) \end{bmatrix} \\
&= \begin{bmatrix} \rho(x) \\ 2\rho(x) \\ \vdots \\ W\rho(x) \end{bmatrix} \\
&= \tilde{A}_2 \circ \rho \circ \tilde{A}_1(x).
\end{aligned}$$

Step 3: (conclusion) Now, we know that all networks Φ_B are unique, since they can be rewritten in a way where we can use theorem 4. Furthermore, we know that the log-cardinality of such networks is $\Omega(W^2)$. Now, since Φ_B are all inside $\mathcal{F}_a^b(\mathcal{A}=\rho,\mathcal{L}=4,\mathcal{W}=W)$, we have that

$$\log \#\mathcal{F}_a^b(\mathcal{A}=\rho,\mathcal{L}=4,\mathcal{W}=W) \geq \log \#B \in \Omega(W^2).$$

We can than simply count the number of parameters of such network to show that

$$\log \#\mathcal{F}_a^b(\mathcal{A}=\rho,\mathcal{L}=4,\mathcal{W}=W) \in \mathcal{O}(W^2).$$

Having both an upper and a lower bound, the proof is concluded □