Upper bounds on the Natarajan dimensions of some function classes

Ying Jin
Department of Statistics, Stanford University
Stanford, CA, United States of America
ying531@stanford.edu

Abstract—The Natarajan dimension is a fundamental tool for characterizing multi-class PAC learnability, generalizing the Vapnik-Chervonenkis (VC) dimension from binary to multi-class classification problems. This work establishes upper bounds on Natarajan dimensions for certain function classes, including (i) multi-class decision tree and random forests, and (ii) multi-class neural networks with binary, linear and ReLU activations. These results may be relevant for describing the performance of certain multi-class learning algorithms.

I. BACKGROUNDS

Many tasks in statistical learning concern finding a good representation of the true relationship underlying the observations out of a perhaps huge family of functions. To this end, an intuitive and prominent approach is empirical risk minimization (ERM). Given i.i.d. observations \( \{(X_i, Y_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y} \) and a loss function \( \ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+ \), to learn a predictor \( f \in \mathcal{F} \) for \( Y \) with the smallest loss, ERM selects the candidate with the smallest empirical prediction error:

\[
\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{L}_n(f), \quad \hat{L}_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(X_i), Y_i).
\]

Here, it is hoped that \( \hat{L}_n(f) \) is a good estimate of the true prediction error \( L(f) = \mathbb{E}[\ell(f(X), Y)] \). The complexity of the function class \( \mathcal{F} \) is a crucial quantity that impacts the performance of such \( \hat{f} \). The more complex \( \mathcal{F} \) is, the more likely it is that the smallest empirical risk occurs only by chance, and that the estimate for \( \hat{f} \) is far from its true value. Understanding the complexity of function classes to characterize the learning performance is a fundamental task in statistical learning theory.

A. Natarajan dimension

This paper studies the Natajaran dimension (1), a measure of the complexity in learning function classes for multi-class classification problems. It generalizes the well-known Vapnik-Chervonenkis (VC) dimension (2) for binary classification function classes. A closely related quantity, the graph dimension, is also defined below.

Definition I.1. Let \( \mathcal{H} \) be a class of functions \( h: \mathcal{X} \rightarrow \mathcal{Y} \). The growth function of \( \mathcal{H} \) is defined as

\[
G(\mathcal{H}, n) := \max_{x_1, \ldots, x_n \in \mathcal{X}} \left| \{(f(x_1), f(x_2), \ldots, f(x_n)): f \in \mathcal{H}\} \right|,
\]

which is the number of distinct realizations of \( f \in \mathcal{H} \) on any \( n \) feature values, where for a finite set \( A \), we write \(|A|\) as the cardinality of \( A \).

More specifically, following Definition I.1, the functions in \( \mathcal{H} \) has at least \( 2^{G(\mathcal{H}, n)} \) different configurations on \( d_N(\mathcal{H}) \) feature values in \( \mathcal{X} \). This fact allows us to utilize \( 2^{d_N(\mathcal{H})} \leq G(\mathcal{H}, d_N(\mathcal{H})) \) to obtain proper upper bounds on \( d_N(\mathcal{H}) \) for the function classes \( \mathcal{H} \) we study in this paper.

We say that \( \mathcal{H} \) \( N \)-shatters \( S \) if there exists \( f_1, f_2: S \rightarrow \mathcal{Y} \) such that \( f_1(x) \neq f_2(x) \) for all \( x \in S \), and for every \( T \subseteq S \), there exists some \( g \in \mathcal{H} \) such that

\[
\forall x \in T, \ g(x) = f_1(x), \quad \text{and} \quad \forall x \in S \setminus T, \ g(x) = f_2(x).
\]

The graph dimension of \( \mathcal{H} \), denoted as \( d_G(\mathcal{H}) \), is the maximal cardinality of any set \( G \)-shattered by \( \mathcal{H} \). The Natarajan dimension of \( \mathcal{H} \), denoted as \( d_N(\mathcal{H}) \), is the maximal cardinality of any set \( N \)-shattered by \( \mathcal{H} \).

Both the graph dimension and the Natarajan dimension coincide with the VC dimension for \( \mathcal{Y} = \{0, 1\} \), and it is shown that \( d_N(\mathcal{H}) \leq d_G(\mathcal{H}) \leq 4.67 \log_2(|\mathcal{Y}|) d_N(\mathcal{H}) \) (3). In this work, we only focus on the Natarajan dimension, and the results on the graph dimension can be easily obtained.

Similar to the VC dimension, the Natarajan dimension can be used to characterize the generalization of ERM – more precisely, the true prediction error of the empirical risk minimizer – using multi-class classification function classes (1; 3; 4). However, unlike the extensively studied VC dimension, results on upper bounds on the Natarajan dimensions are relatively rare; existing results only cover linear function classes and reduction tree classes (4). This work provides two more instances, decision trees (and random forests) and neural networks (fully-connected ones with linear, binary and ReLU activations). We study tree-based function classes in Section II and neural networks in Section III, with all proofs in Section IV.

B. Growth functions

Our theoretical proof will follow the classical idea of bounding growth functions.

Definition I.2. Let \( \mathcal{H} \) be a class of functions \( h: \mathcal{X} \rightarrow \mathcal{Y} \). The growth function of \( \mathcal{H} \) is defined as

\[
G(\mathcal{H}, n) := \max_{x_1, \ldots, x_n \in \mathcal{X}} \left| \{(f(x_1), f(x_2), \ldots, f(x_n)): f \in \mathcal{H}\} \right|,
\]

which is the number of distinct realizations of \( f \in \mathcal{H} \) on any \( n \) feature values, where for a finite set \( A \), we write \(|A|\) as the cardinality of \( A \).
C. Related work

The Natarajan dimension and the graph dimension in Definition I.1 are both proposed by (1) that generalize the Vapnik-Chervonenkis (VC) dimension (2) to multi-class problems. The Natarajan dimension is a crucial measure of PAC learnability; (4) shows that the sample complexity of PAC learning for multi-class classifications are bounded, in both directions, in terms of the Natarajan and the graph dimensions.

This work is closely related to the literature on establishing bounds on the Natarajan dimensions of function classes, including (4) on generalized linear models and reduction trees, (5) on multi-class support vector machines, (6) on one-versus-all, all-pairs, error-correcting-output-codes methods, etc. This paper extends this line of work by establishing new bounds on popular tree-based and neural-network-based function classes. Notably, the decision trees we consider differ from the reduction trees in (4), and should be viewed as distinct function classes; see Remark II.1 for a detailed comparison.

The proof ideas in the work are inspired by several works in bounding the VC dimensions of binary-classification function classes. In particular, the technique in proving the results for neural network classifiers in Section III extends (7). The author was not aware of previous work on upper bounding the VC dimensions on decision trees with a given number of real-valued features and a given depth before a preliminary version of this work (as arXiv preprint 2209.07015). As discussed in an independent work (8), such results appear to be very recent.

II. TREE-BASED FUNCTION CLASS

Decision trees and random forests are popular tree-based machine learning methods that could be used for multi-class classification. This section provides upper bounds on the Natarajan dimensions of these classes.

We first study the function class $\Pi_{L,d}$ of decision trees with a depth-$L$ $\ell$-class decision tree. A depth-$L$ decision tree is a full binary tree, where each internal node $v$ is associated with a feature $i_v \in \{1, \ldots, p\}$ and a threshold $\theta_v \in \mathbb{R}$, and each leaf node is associated with a class $k \in \{1, \ldots, d\}$. For input $x \in \mathbb{R}^p$, the output is obtained by traversing a path of length $L - 1$ from the root node to the leaf node. At each node $v$, if $x_{i_v} \leq \theta_v$ then we continue to its left child node, and to its right child node otherwise. The final classification is given by the class associated with the leaf node we arrive at.

Remark II.1. (6) also studies the Natarajan dimension for decision trees, however, under a different definition: they assume there is a bijection between the leaf nodes and the $d$ classes, and the internal nodes are from a general class of binary functions. Instead, we allow multiple leaf nodes to represent the same class, but consider more restricted binary classification rules for the internal nodes. We will obtain different upper bounds with different proof techniques.

The following theorem establishes upper bounds on the Natarajan dimension of $\Pi_{L,d}$.

Theorem II.2. The Natarajan dimension of $\Pi_{L,d}$ with inputs from $\mathbb{R}^p$ is no greater than $O(L^2 \log(pd))$.

We then consider the function class of random forests, denoted by $\Pi_{L,T,d}$, each element of which is a random forest classifier $F(\cdot)$ consisting of $T$ depth-$L$ $d$-class decision trees $f_j(\cdot), j = 1, \ldots, T$. Given any $x \in \mathbb{R}^p$, the output of a random forest is given by $F(x) = \text{argmax}_{1 \leq k \leq d} \sum_{j=1}^T I\{f_j(x) = k\}$, the most-frequently predicted class among all $T$ trees. Its Natarajan dimension can be bounded as follows.

Theorem II.3. The Natarajan dimension of $\Pi_{L,T,d}$ with inputs from $\mathbb{R}^p$ is no greater than $O(LT^2 \log(pd))$.

As we discussed in Section I-C, upper bounds on the VC dimension of decision trees with real-valued features appear to be very recent results (8). In particular, the independent recent work of (8) shows that binary decision trees with $L_T$ leaves for $p$ real-valued features and the same splitting rules as ours has a VC dimension of $O(L_T \log(L_T))$. Noting that the number of leaves is $2^L - 1$ for a tree of depth $L$, our bound $O(L^2 \log(pd))$ in Theorem II.2 only pays a price of $\log(d)$ for $d$-class classification compared with the results in (8), while the rate in the number of leaves is the same.

III. NEURAL NETWORK FUNCTION CLASS

For a number of $d$ actions, a multiple classification neural network has $d$ outputs in the final layer and constructs a classification by taking the maximum over these outputs.

A. Neural network function class with binary and linear activations

We first consider $\Pi_p^{\text{binary}}_R$, a neural network function class with a fixed structure $S$ of $p$ parameters, where all activation functions are either binary or linear, which generalizes the setting in (7).

The fixed structure $S$ consists of $L$ layers, where the $\ell$-th layer has $n_{\ell}$ nodes, $\ell \in \{1, \ldots, L\}$. We denote the $j$-th node in layer $\ell$ as $\text{Node}_{\ell,j}$, and denote $\mathcal{N}_{\ell,j}$ as the set of nodes in layer $\ell - 1$ that are connected to $\text{Node}_{\ell,j}$, whose size is $m_{\ell,j} = |\mathcal{N}_{\ell,j}|$. There is one real-valued parameter for each pair of connected nodes in adjacent layers. We define the set of parameters of the network as $w = \{w_{\ell,j,s}\}_{1 \leq s \leq m_{\ell,j}, 1 \leq \ell \leq L}$ (excluding those weights for the last output layer), which we assume is of a size smaller than $p$.

Each element in $\Pi_p^{\text{binary}}$ is a feed-forward neural network; given any input $x \in \mathbb{R}^n$, it outputs $f(x; w)$ as follows. The input layer takes $x \in \mathbb{R}^m$ from $m$ input nodes, each for one feature. In each hidden layer $\ell \in \{1, \ldots, L - 1\}$, $\text{Node}_{\ell,j}$ performs a linear combination to the outputs from each node in $\mathcal{N}_{\ell,j}$ in the previous layer, using $m_{\ell,j}$ parameters $w_{\ell,j,1}, \ldots, w_{\ell,j,m_{\ell,j}} \in \mathbb{R}$. To be specific, the output of $\text{Node}_{\ell,j}$ is $f_{j,\ell}(x) = \sigma \left( \sum_{s \in \mathcal{N}_{\ell,j}} w_{\ell,j,s} \cdot f_{s}^{\ell-1}(x) \right)$, where $\sigma(\cdot)$ is either the binary activation $\sigma(z) = 1\{z > 0\}$ or the linear activation $\sigma(z) = z$ for $z \in \mathbb{R}$. The last layer has $d$
nodes, and is fully-connected to the last hidden layer with an identity activation. The output of the neural network is
\[ f(x; w) = \arg\max_{1 \leq k \leq d} \left\{ \sum_{s=1}^{k-1} w_{L,k,s} \cdot f_{n}^{(L-1)}(x) \right\}. \]

The following theorem provides upper bounds on Natarajan dimensions of such function classes.

**Theorem III.1.** The Natarajan dimension of \( \Pi_{p,S}^{\text{binary}} \) described above is upper bounded by \( O(d \cdot p^2) \).

Several seminal early works have established the VC dimension of neural networks. The textbook result in (9) provides an upper bound of \( O(p \log p) \) for neural networks with \( p \) parameters and all binary activation functions \( \sigma(z) = 1 \{z > 0\} \). (7) provides an upper bound of \( O(p^2) \) for neural networks with \( p \) parameters and binary or linear activation functions as considered in this subsection. By comparing this two results, we see that the allowing for linear activation functions adds a factor of \( p \) to the VC dimension (from the rate of \( p \log p \) to \( p^2 \)).

Also, moving from binary to \( d \)-class classification adds a multiplicative factor of \( d \) to the upper bound when comparing Theorem III.1 with (7). This is because the \( d \)-output neural network structure leads to a power-\( d \) factor in the growth function using our current proof technique. It will be interesting to see whether such dependence on \( d \) can be further sharpened.

**B. Neural network function class with ReLU activations**

We now consider \( \Pi_{p,S}^{\text{ReLU}} \), the class of multiple classification neural networks with a given structure \( S \), which contains at most \( p \) parameters in intermediate layers and \( d \) final outputs, and the activation functions are either binary, linear or ReLU, i.e., \( \sigma(z) = z \) or \( \sigma(z) = 1 \{z > 0\} \) or \( \sigma(z) = z 1 \{z > 0\} \). The definition of structure \( S \) is the same as in the preceding subsection; the only difference is that the activation functions for internal nodes can now be more general.

The Natarajan dimension of this function class is upper bounded as in the following theorem.

**Theorem III.2.** The Natarajan dimension of \( \Pi_{p,S}^{\text{ReLU}} \) described above is upper bounded by \( O(d \cdot p^2) \).

Our bound in Theorem III.2 is of the same order as that in Theorem III.2. Intuitively, this is because the ReLU activation can be viewed as a combination of binary and linear activation, which does not significantly increase the growth function compared with networks with the latter two activation functions.

Our theoretical analysis for the Natarajan dimensions of neural network function class is largely inspired by the framework of (7), which expresses neural network outputs as depending on linear combinations of binary values and original features, where the linear coefficients are further polynomials of the parameters of a bounded degree.

**Remark III.3.** The price we pay for the number of classes \( d \) in neural networks is a multiplicative factor of \( d \), which is higher than the \( \log(d) \) factor for decision trees and random forests. Although we only provide an upper-bound analysis, such comparison potentially indicates that the complexity of multi-class classification with neural networks might be higher than that of tree-based function classes.

**IV. TECHNICAL PROOFS**

**A. Proof of Theorem II.2**

Given a set of inputs \( \{x_1, \ldots, x_n\} \in \mathbb{R}^p \), we first bound the number of configurations of the output on \( \{x_1, \ldots, x_n\} \) by a decision tree of depth \( L \). Denote the growth function on \( x_1, \ldots, x_n \) as \( g(\Pi_{\text{tree}}^d, n \mid x_1, \ldots, x_n) = \{(f(x_1), f(x_2), \ldots, f(x_n)) \mid f \in \Pi_{\text{tree}}^d \} \). We sort and denote internal nodes as \( v \in \{1, \ldots, V\} \), where \( V = 2^{L-1} - 1 \) is the total number of internal nodes. We denote the corresponding feature as \( i_v \), and the threshold as \( \theta_v \), which vary with the tree \( f \in \Pi_{\text{tree}}^d \). We index the set of leaf nodes with \( \{1, \ldots, 2^{L-1}\} \), and each leaf node \( l \leq 1 \) represents a class \( k_l \in \{1, \ldots, d\} \). In this way, given any \( x_i \in \mathbb{R} \), the output from a tree \( f \in \Pi_{\text{tree}}^d \) is determined by the vector of queries \( (1 \{x_{i,v} \leq \theta_v\})_{v=1}^V \in \mathbb{R}^V \) at all the \( V \) internal nodes, as well as \( k_1, 1 \leq l \leq 2^{L-1} \), the assignment of classes to all leaf nodes. To better represent the classification model, for a decision tree \( f \in \Pi_{\text{tree}}^d \), we let \( I_{f,v}(x) = 1 \{x_{i,v} \leq \theta_v\} \) be the query function at node \( v \), \( I_{f}(x) = (1 \{x_{i,v} \leq \theta_v\})_{v=1}^V \) be the vector of queries at internal nodes, and \( L(f) = (k_l)_{l=1}^V \) be the vector of leaf node assignments.

Since \( f(x) \) is fully decided by \( I_{f}(x) \) and \( L(f) \), we obtain an upper bound on the growth function that
\[
g(\Pi_{\text{tree}}^d, n \mid x_1, \ldots, x_n) \leq \left| \{(I_{f}(x_1), I_{f}(x_2), \ldots, I_{f}(x_n), L(f)) \mid \right. \]
\[
\left. (f(x_1), f(x_2), \ldots, f(x_n)) \in \{0, 1\}^{n \times V} : f \in \Pi_{\text{tree}}^d \} \right| \]
\[
\leq \left( \left| \{(I_{f}(x_1), I_{f}(x_2), \ldots, I_{f}(x_n)) \in \{0, 1\}^{n \times V} : f \in \Pi_{\text{tree}}^d \} \right| \right. \]
\[
\left. \times (L(f) \in \{1, \ldots, d\}^{2^{L-1}} : f \in \Pi_{\text{tree}}^d \} \right| \]

The second term is upper bounded by the cardinality of the image as \( \left| \{(f(x_1), I(x_2), \ldots, I(x_n)) \in \{0, 1\}^n : f \in \Pi_{\text{tree}}^d \} \right| \leq d^{2L-1} \). For the first term, the total number of different configurations of \( I_{f} \) is upper bounded by the product of those of all \( I_{f,v} \), namely,
\[
\prod_{v=1}^V \left| \{(I_{f,v}(x_1), I_{f,v}(x_2), \ldots, I_{f,v}(x_n)) \in \{0, 1\}^n : f \in \Pi_{\text{tree}}^d \} \right| \]
\[
\leq \prod_{v=1}^V \left| \{(I_{f,v}(x_1), I_{f,v}(x_2), \ldots, I_{f,v}(x_n)) \in \{0, 1\}^n : f \in \Pi_{\text{tree}}^d \} \right| \]
\[
= \left| \{(f(x_1), I(x_2), \ldots, I(x_n)) \in \{0, 1\}^n : f \in \Pi_{\text{tree}}^d \} \right| \]

since each query function \( I_{f,v} \) belongs to the function class
\[
\mathcal{I} = \{ I : \mathbb{R}^p \to \{0, 1\} : I(x) = 1 \{x_i \leq \theta\}, \ i \in \{1, \ldots, p\}, \ \theta \in \mathbb{R} \}.
\]
Moreover, for any function $I \in \mathcal{I}$ of the form $I(x) = 1 \{x_s \leq \theta\}$ for some feature $s \in \{1, \ldots, p\}$, there are at most $n + 1$ different classifications on the $n$ samples: one could sort all possible classification results into a sequence of size $n + 1$ (perhaps with recurring members), such that the $\ell$-th of result classifies the $\ell$ sample with smallest $x_s$ as positive and others as negative. Hence
\[
\left\{(I(x_1), I(x_2), \ldots, I(x_n)) \in \{0, 1\}^n : I \in \mathcal{I}\right\} \leq p(n + 1).
\]
Putting this together, we have $g(\Pi^{\text{forest}}_{L,d}, n \{x_1, \ldots, x_n\}) \leq (p(n + 1))^{2^{L-1}-1} \cdot d^{2^{L-1}}$. Now suppose $\{x_1, \ldots, x_N\}$ are $N$-shattered by $\Pi^{\text{forest}}_{L,d}$. By definition, policies in $\Pi^{\text{forest}}_{L,d}$ have at least $2^N$ different configurations on $\{x_1, \ldots, x_N\}$, leading to
\[
2^N \leq (p(n + 1))^{2^{L-1}-1} \cdot d^{2^{L-1}}.
\]
Taking logarithm of both sides yields $N \log 2 \leq 2L \log(pd) + 2L \log N$, which further gives $N = O(L^2 \log(pd))$. This proves the upper bound of Natarajan dimension of $\Pi^{\text{forest}}_{L,d}$.

### B. Proof of Theorem II.3

Suppose $\{x_1, \ldots, x_N\}$ is $N$-shattered by $\Pi^{\text{forest}}_{L,T,d}$. Since the output of a policy $f \in \Pi^{\text{forest}}_{L,T,d}$ is fully decided by the outputs of $T$ decision trees $f_1, \ldots, f_T \in \Pi^{\text{tree}}_{L,d,j}$, the number of different configurations on $\{x_1, \ldots, x_N\}$ by policies in $\Pi^{\text{forest}}_{L,T,d}$ is upper bounded by $g(\Pi^{\text{tree}}_{L,d}, N \{x_1, \ldots, x_N\})^T$. Hence,
\[
2^N \leq g(\Pi^{\text{tree}}_{L,d}, N)^T \leq (p(N + 1))^{2^{L-1}-1} \cdot d^{2^{L-1}}.
\]
This implies $N \leq O(LT^2 \log(pdT))$, which completes the proof of Theorem II.2.

### C. Proof of Theorem III.1

Firstly, we fix any $f \in \Pi^{\text{binary}}_{L,s,k}$. Note that the final output $f(x)$ is fully determined by the $(d(d - 1))/2$ binary functions $\{b_{k,k'}(\cdot) : 1 \leq k < k' \leq d\}$, where
\[
b_{k,k'}(x) = \mathbb{I} \left( \sum_{s=1}^{n-1} \theta_{L,k,s} f_s^{(L-1)}(x) - \sum_{s=1}^{n-1} \theta_{L,k',s} f_s^{(L-1)}(x) \right)
\]
for $I(z) = \mathbb{I} \{z > 0\}$. For simplicity, we assume there are no ties among the $k$ outputs; otherwise one could break the ties without changing our results.

We now proceed to analyze the outputs $f^{(j)}(x)$ in the hidden layers for $\ell \leq L - 1$. Similar to the idea in (7), we will show that they can all be expressed as linear combinations of binary values and $x_1, \ldots, x_m$. To begin with, note that the activation functions $\sigma(\cdot)$ are either binary $\sigma(z) = \mathbb{I} \{z > 0\}$ or linear $\sigma(z) = z$, hence the outputs at layer $\ell = 1$ are either binary-valued, or some function of $x$. That is, we either have $f^{(1)}(x) \in \{0, 1\}$, or $f^{(1)}(x) = x^\top \alpha_1^{(1)}$ for some $\alpha_1^{(1)} \in \mathbb{R}^m$. We let $f^{(1)}(x) = \sigma(g^{(1)}_s(x))$, i.e., we denote $g^{(1)}_s(x)$ as the intermediate output before applying the activation function for node $s$ at layer $j$. By definition, $g^{(1)}_s(x)$ is some linear combination of $f^{(1)}(x)$, the node outputs at layer 1. Therefore, for any $s$, $g^{(2)}_s(x)$ is a linear combination of some binary values and some linear function of $x$. Letting $I(z) = \mathbb{I} \{z > 0\}$. We can write
\[
g^{(2)}_s(x) = \sum_{j \in N_{s,j}, \sigma(z) = 1 \{z > 0\}} w_{2,s,j} x^\top \alpha_1^{(1)} + \sum_{j \in N_{s,j}, \sigma(z) = z} w_{2,s,j} x^\top \alpha_1^{(1)}
\]
\[
= \sum_{j \in N_{s,j}, \sigma(z) = 1 \{z > 0\}} w_{2,s,j} (\theta_{1,j,1} x_1 + \theta_{1,j,2} x_2 + \cdots + \theta_{1,j,m} x_m)
\]
\[
+ \sum_{j \in N_{s,j}, \sigma(z) = z} \theta_{2,s,j} (\theta_{1,j,1} x_1 + \theta_{1,j,2} x_2 + \cdots + \theta_{1,j,m} x_m)
\]
for some $\theta_{1,j,1}, \ldots, \theta_{1,j,m} \in \mathbb{R}$ that can be derived from the network structure and $\{w_{j,s,n}\}$, where $n_j$ are the total number of $\theta_{2,s,n}$, needed in such expression. Following this rule, we can write
\[
g^{(3)}_s(x) = \sum_{j \in N_{s,j}, \sigma(z) = 1 \{z > 0\}} \theta_{2,s,n,j} \alpha_j^{(1)}
\]
\[
g^{(2)}_s(x) \text{ at this node is } \sigma(z) = 1 \{z > 0\}
\]
\[
+ \sum_{j \in N_{s,j}, \sigma(z) = z} \alpha_j^{(1)}
\]
\[
\text{other } g^{(2)}_s \text{ with activation function } \sigma(z) = 1 \{z > 0\}
\]
\[
+ \cdots + \cdots + \cdots + \cdots
\]
\[
\text{other } g^{(2)}_s \text{ with activation function } \sigma(z) = z
\]
\[
\sum_{s \in I \{\cdots\}} \theta_{2,s,n,s} (\theta_{2,1,n_1,1} x_1 + \cdots + \theta_{2,1,n_1,m} x_m) + \cdots + \cdots + \cdots + \cdots
\]

where $\theta_{2,1,n_1,1}, \ldots, \theta_{2,1,n_1,m} \in \mathbb{R}$ and $\theta_{2,1,n_2,1}, \ldots, \theta_{2,1,n_2,m} \in \mathbb{R}$ are such coefficients. In the last expression, each $I(\cdots)$ represents a node with binary activation, and inside the argument is another linear combination of binary values and $x_1, \ldots, x_m$. So on and so forth, we have
\[
g^{(L-1)}_s(x) = \theta_{L-1,s,n_1} \sum_{n_1=1}^{n_1} \theta_{L-2,n_1,1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-2,n_m,1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-2,n_m,1} \cdots
\]
\[
+ \theta_{L-1,s,n_1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-1,n_m,1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-1,n_m,1} \cdots
\]
\[
+ \theta_{L-1,s,n_1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-1,n_m,1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-1,n_m,1} \cdots
\]
\[
+ \theta_{L-1,s,n_1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-1,n_m,1} \cdots \sum_{n_m=1}^{n_m} \theta_{L-1,n_m,1} \cdots
\]
for some $\theta_{L-1,s,n_1,1}, \ldots, \theta_{L-1,s,n_m,1} \in \mathbb{R}$ and $\theta_{L-1,s,n_1,1}, \ldots, \theta_{L-1,s,n_m,1} \in \mathbb{R}$. Put another way, the outputs $g^{(L-1)}_s(x)$ can be written as the linear combination of outputs of some binary functions and original features, and the arguments to these binary functions are again some linear combination of binary functions (computed from preceding layers) and the original features, and so on. Also, the coefficients $\theta_{j,s,n}$ are all polynomials of $x_1, \ldots, x_m$, $w_1, \ldots, w_p$, whose degrees are no larger than $p$ since there are at most $p$ layers. Similarly, the function $b_{k,k'}(x)$ can be
written as a binary function $b_{k,K}(x) = I(c_{k,K}(x))$, where $c_{k,K}(x)$ is a linear combination of several binary functions decided by previous layers and the original features, for which the linear coefficients are polynomials of $x_1, \ldots, x_m$, the original parameters $w_1, \ldots, w_p$ and $\{w_{L,k,s}\}_{1 \leq k \leq d, 1 \leq s \leq n_L-1}$ of degree $\leq p + 1$.

Assume there are $p_1$ binary nodes in intermediate layers. Then those binary functions have at most $2^{p_1}$ configurations. Therefore, for any input $x, x' \in \mathbb{R}^m$ and parameters $w, w' \in \mathbb{R}^p$, we have $f(x; w) = f(x'; w')$ if the set of binary functions

$$T := \left\{ I(\theta_{t,j,i} b_1 + \cdots + \theta_{t,j,p} b_{p_1} + \theta_{t,j,n_{t-1}} x_1 + \cdots + \theta_{t,j,n_t} x_m) : (b_1, \ldots, b_{p_1}) \in \{0, 1\}^{p_1}, 1 \leq \ell \leq L, 1 \leq j \leq n_t \right\}$$

take the same value, where $\theta_{t,j,s}$ are fixed polynomials of the parameters $w_1, \ldots, w_p$, features $x_1, \ldots, x_m$ and $\{w_{L,k,s}\}_{1 \leq k \leq d, 1 \leq s \leq n_L-1}$ of degree $\leq p + 1$. Note that there are at most $(d^2 + p)2^p$ functions in $T$, because for each configuration of $(b_1, \ldots, b_{p_1})$, there are at most $p$ binary nodes in the intermediate layers, and no more than $d^2$ comparisons among the $d$ classes in the last layer. In other words, $T = \left\{ I(P_{t}(x, w_1, \ldots, w_p, \{w_{L,k,s}\}_{1 \leq k \leq d, 1 \leq s \leq n_L-1})) : r = 1, \ldots, R \right\}$, where $R \leq (p + d^2)2^p$ and each $P_{t}(\cdot)$ is a polynomial with degree $\leq p + 1$. If we view the input $x$ as fixed, the above set of functions is

$$T(x) := \left\{ I(P_{t}(x, w_1, \ldots, w_p, \{w_{L,k,s}\}_{1 \leq k \leq d, 1 \leq s \leq n_L-1})) : r = 1, \ldots, R \right\},$$

and each $P_{t}(\cdot)$ is a polynomial of degree $\leq p + 1$ of no greater than $p(1 + d)$ variables

$$\{w_1, \ldots, w_p, \{w_{L,k,s}\}_{1 \leq k \leq d, 1 \leq s \leq n_L-1}\},$$

since the parameters for the last layer is at most $p \cdot d$.

Let $x_1, \ldots, x_N$ be N-shattered by the function class $\Pi^\text{binary}_{p,s}$, and let $w^{(1)}, \ldots, w^{(M)}, M = 2^N$ be the parameters that witness the shattering. Consider the classifications

$$\begin{pmatrix}
  f(x_1; w^{(1)}), & f(x_2; w^{(1)}), & \cdots, & f(x_N; w^{(1)}) \\
  f(x_1; w^{(2)}), & f(x_2; w^{(2)}), & \cdots, & f(x_N; w^{(2)}) \\
  \vdots & \vdots & \ddots & \vdots \\
  f(x_1; w^{(M)}), & f(x_2; w^{(M)}), & \cdots, & f(x_N; w^{(M)})
\end{pmatrix}$$

By the definition of N-shattering, every two rows in the above matrix are distinct. Thus for each $j \neq j'$, there exists some $i \in \{1, \ldots, N\}$ such that $f(x_i; w^{(j)}) \neq f(x_i; w^{(j')})$. By above arguments, there exists some binary function in $T$ that takes different values on $(x_i, w^{(j)})$ and $(x_i, w^{(j')})$. Then there must exist some $1 \leq r \leq R$ such that the signs of $P_{t_1}(x_i, w^{(j)})$ and $P_{t_2}(x_i, w^{(j')})$ are different. In other words, each $j \in \{1, \ldots, M\}$ gives a unique configuration of the signs of the $N \cdot R$ polynomials in $T(x_1), \ldots, T(x_N)$. The following lemma establishes an upper bound for the number of such configurations, which is a re-statement of Corollary 2.1 in (10).

**Lemma IV.1.** Let $\{P_1, \ldots, P_M\}$ be $R$ polynomials of degree at most $\tilde{p}$ in $\tilde{n}$ real variables with $\tilde{R} \geq \tilde{n}$, then the number of different configurations of signs to the $\{P_1, \ldots, P_M\}$ is at most $(8e\tilde{R}\tilde{n})^M$.

Utilizing Lemma IV.1 with $\tilde{R} = NR$, $\tilde{p} = p + 1$ and $\tilde{n} = p(1 + d)$, the number of different configurations on the $N \cdot R$ polynomials is upper bounded as

$$M = 2^N \leq \left( \frac{8e(p + 1) \cdot N(p + d^2) \cdot 2^p}{p(1 + d)} \right)^{p(1 + d)}.$$
REFERENCES

[1] B. K. Natarajan, “On learning sets and functions,” Machine Learning, vol. 4, no. 1, pp. 67–97, 1989.

[2] V. N. Vapnik and A. Y. Chervonenkis, “On the uniform convergence of relative frequencies of events to their probabilities,” in Measures of complexity. Springer, 2015, pp. 11–30.

[3] S. Ben-David, N. Cesa-Bianchi, and P. M. Long, “Characterizations of learnability for classes of {O,, n}-valued functions,” in Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 333–340.

[4] A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz, “Multiclass learnability and the erm principle,” in Proceedings of the 24th Annual Conference on Learning Theory. JMLR Workshop and Conference Proceedings, 2011, pp. 207–232.

[5] Y. Guermeur, “Sample complexity of classifiers taking values in q, application to multi-class svms,” Communications in Statistics Theory and Methods, vol. 39, no. 3, pp. 543–557, 2010.

[6] A. Daniely, S. Sabato, and S. Shwartz, “Multiclass learning approaches: A theoretical comparison with implications,” Advances in Neural Information Processing Systems, vol. 25, 2012.

[7] E. D. Sontag et al., “Vc dimension of neural networks,” NATO ASI Series F Computer and Systems Sciences, vol. 168, pp. 69–96, 1998.

[8] J.-S. Leboeuf, F. LeBlanc, and M. Marchand, “Generalization properties of decision trees on real-valued and categorical features,” arXiv preprint arXiv:2210.10781, 2022.

[9] S. Shalev-Shwartz and S. Ben-David, Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.

[10] P. W. Goldberg and M. R. Jerrum, “Bounding the vplchervonenkis dimension of concept classes parameterized by real numbers,” Machine Learning, vol. 18, no. 2-3, pp. 131–148, 1995.