Universal approximations of permutation invariant/equivariant functions by deep neural networks

Akiyoshi Sannai\textsuperscript{1}, Yuuki Takai\textsuperscript{2,}, Matthieu Cordonnier\textsuperscript{3}

\textsuperscript{1,2}RIKEN Center for Advanced Intelligence Project, Tokyo, Japan, Keio University, Kanagawa, Japan
\textsuperscript{3}École Normale Supérieure Paris-Saclay, Cachan, France

\textsuperscript{1}akiyoshi.sannai@riken.jp, \textsuperscript{2}yuuki.takai@riken.jp, \textsuperscript{3}matthieu.cordonnier@ens-paris-saclay.fr

Abstract

In this paper, we develop a theory of the relationship between permutation (\(S_n\)) invariant/equivariant functions and deep neural networks. As a result, we prove a permutation invariant/equivariant version of the universal approximation theorem, i.e \(S_n\)-invariant/equivariant deep neural networks. The equivariant models are consist of stacking standard single-layer neural networks \(Z_i: X \to Y\) for which every \(Z_i\) is \(S_n\)-equivariant with respect to the actions of \(S_n\). The invariant models are consist of stacking equivariant models and standard single-layer neural networks \(Z_i: X \to Y\) for which every \(Z_i\) is \(S_n\)-invariant with respect to the actions of \(S_n\). These are universal approximators to \(S_n\)-invariant/equivariant functions. The above notation is mathematically natural generalization of the models in [Zaheer et al., 2018]. We also calculate the number of free parameters appeared in these models. As a result, the number of free parameters appeared in these models is much smaller than the one of the usual models. Hence, we conclude that although the free parameters of the invariant/equivariant models are exponentially fewer than the one of the usual models, the invariant/equivariant models can approximate the invariant/equivariant functions to arbitrary accuracy. This gives us an understanding of why the invariant/equivariant models designed in [Zaheer et al., 2018] work well.

1 Introduction

Deep neural networks have great success in many applications such as image recognition, speech recognition, natural language process and others [Alex et al., 2012], [Goodfellow et al., 2013], [Wan et al., 2013], [Silver et al., 2017]. A point common in their works is that they construct larger and deeper networks. When we use larger and deeper networks, one of the main obstructions to learn is so called the curse of dimensionality, i.e. if the parameter, dimension of the models increase, then the required sample size and hence computational complexity for approximation increases exponentially.

One idea to overcome the curse of dimensionality is to design models which fit to objectives.

[Zaheer et al., 2018] designed a model for machine learning tasks defined on sets, which are mathematically, permutation invariant/equivariant tasks. They demonstrate surprisingly good applicability on their method on population statistic estimation, point cloud classification, set expansion, and outlier detection. Empirically speaking, their result is no doubt meaningful. However, its theoretical guarantee is not considered sufficiently in their paper. Hence, because of this motive, we generalize their models and give invariant/equivariant versions of the universal approximation theorem in this paper. Our theorem is not only for the classical depth-bounded version but also the width-bounded version.

Furthermore, we also calculate the number of free parameters appeared in our invariant/equivariant models. As a result, the number of free parameters appeared in our models is exponentially smaller than the one of the usual models. Hence, we conclude that although the free parameters of the invariant/equivariant models are exponentially fewer than the one of the usual models, the invariant/equivariant models can approximate the invariant/equivariant functions to arbitrary accuracy. This gives us an understanding of why the invariant/equivariant models designed in [Zaheer et al., 2018] work well.

The universal approximation theorem, which is proved by [Cybenko, 1989] first in the literature. His theorem states that when the width goes to infinity, a (usual) neural network with a single hidden layer can approximate any continuous function with compact support to arbitrary accuracy. Though his theorem was only for sigmoid activation functions, there are versions of the universal approximation theorems including the version for ReLU activation functions [Barron, 1994], [Hornik et al., 1989], [Funahashi, 1989], [Kůrková and Sanguineti, 2002], [Sonoda and Murata, 2017].

The advantage of the universal approximation theorem in learning theory is to guarantee that we can search in the space which contains the solutions. The universal approximation theorem states the existence of the model which approximates the target function in arbitrary accuracy. This means that if we use the suitable algorithm, we have the desired solutions. We cannot guarantee such situations without the universal approximation theorem. Hence, our invariant/equivariant ver-
sions of the universal approximation theorem is important.

A technical key point of the proof is the one to one correspondence between $S_n$-equivariant functions and $\text{Stab}(1)$-invariant functions. Here, $\text{Stab}(1)$ is the subgroup of $S_n$ consisting of the elements which fix $\{1\}$ as permutation. We first confirm this correspondence at the function level, after that, we lift it to deep neural networks. This correspondence enables us to reduce the equivariant case to the invariant case. Another key point is Kolmogorov-Arnold's representation theorem which is famous as the Hilbert's 13th problem [Kolmogorov, 1956], [Arnold, 1957]. This theorem gives us a concrete description of invariant functions. Due to this theorem and the usual universal approximation theorem, we can construct a concrete deep neural networks of the invariant model. Combining with the reduction to the invariant case, we complete the proof.

There are two important invariants in deep neural networks. One is the width and the other is the depth. Using these invariants, Cybenko's universal approximation theorem can be viewed as depth-bounded universal approximation theorem. Another way, width-bounded universal approximation theorem is proved by [Hanin and Sellke, 2018], which states even if the maximal width of deep neural networks is bounded, if the minimal width is bigger than the input dimension, then the universal approximation theorem holds. This kind of the universal approximation theorem is also important from the point of view of expressive power. We give both of invariant/equivariant versions of depth-bounded universal approximation theorem and width-bounded universal approximation theorem in this paper.

1.1 Contributions

Our contributions are summarized as follows:

- We prove the invariant/equivariant versions of the approximation theorem, which is the first step to understand the behavior of deep neural networks with perturbations or more generally group actions. Furthermore, we prove the width-bounded versions of the invariant/equivariant universal approximation theorem.
- We calculate the number of free parameters appeared in our models. As a result, the number of parameter in our models is exponentially smaller than the one of the usual models. This means that our models are easier to train than the usual models.
- As an application of our result, these two results above give an understanding of the models designed in [Zaheer et al., 2018], i.e our approximation theorem and the calculation of the free parameters are evidences for the models in [Hanin and Sellke, 2018] to work well.

1.2 Related works

Group theory, or symmetry is an important concept in mathematics, physics, and machine learning. In machine learning, deep symmetry networks (symnets) is designed in [Gens and Domingos, 2014] as a generalization of convnets that forms feature maps over arbitrary symmetry groups. Group equivariant Convolutional Neural Networks (G-CNNs) is designed in [Cohen and Welling, 2016], as a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symmetries. The models for permutation invariant/equivariant tasks are designed in [Zaheer et al., 2018] to give great results on population statistic estimation, point cloud classification, set expansion, and outlier detection.

The approximation theorem is one of the most classical mathematical theorem of neural networks. The universal approximation theorem states that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of $\mathbb{R}^n$. As we see in the introduction, Cybenko proved this theorem in 1989 for sigmoid activation functions [Cybenko, 1989]. After his achievement, some researchers showed similar results to generalize the sigmoid function to a large class of activation functions [Barron, 1994], [Hornik et al., 1989], [Funahashi, 1989], [Kurkova, 1992]. Their result means that universal approximation is caused by the structure of neural networks.

Recently, with much success of deep learning, the interest in the expressive power increased. Depth efficiency is one of the main problems in expressive power. Though feed-forward networks with a single hidden layer are universal approximators, they need exponentially many neurons for approximation. There exists a family of functions which can be represented much more efficiently with deep networks than with shallow ones as well [Delalleau and Bengio, 2011]. There exists a 3-layer network, which cannot be realized by any 2-layer to more than a constant accuracy if the size grows subexponentially in the dimension [Eldan and Shamir, 2016]. There exists classes of deep convolutional ReLU networks that cannot be realized by shallow ones if the size is no more than an exponential bound [Cohen et al., 2016]. A width-bounded version of the universal approximation theorem is proved in [Lu et al., 2017]. They showed that if the width is bigger than $n + 3$, where $n$ is the input dimension, deep neural networks with ReLU activation functions are universal approximators. Hanin generalized their result, i.e if the width is bigger than the input dimension, neural networks are universal approximators [Hanin and Sellke, 2018].

2 Preliminary

In this paper, we treat fully connected deep neural networks. We basically use ReLU activation functions. The ReLU activation function is defined by

$$\text{ReLU}(x) = \max(0, x) = [x]_+.$$  

However, as we see later, if the usual universal approximation theorem holds, then the proof works [Barron, 1994; Hornik et al., 1989; Funahashi, 1989; Kurkova]. Deep neural networks is built by stacking the blocks which consist of a linear map and a ReLU activation. More formally, it is a function $Z_i$ from $\mathbb{R}^{d_i}$ to $\mathbb{R}^{d_{i+1}}$ defined by $Z_i(x) = \text{ReLU}(W_i x)$, where $W_i \in \mathbb{R}^{d_{i+1} \times d_i}$. In this case, $d_i$ is called the width of the $i$-th layer. The output of the deep neural networks is:

$$Y(x) = Z_H \circ Z_{H-1} \circ \ldots \circ Z_2 \circ Z_1(x),$$

where $H$ is called the depth of the deep neural network. Our models, which are generalization of the models in
[Zaheer et al., 2018], are defined by the invariant/equivariant property of $\mathbb{Z}_n$. Before defining the invariant/equivariant models, we review a formal definition of the action on sets.

**Definition 2.1.** Let $S_n$ be the group of permutations of $n$ elements. Let $X$ and $Y$ be set and $S_n$ having action on $X$ (resp. $Y$) by mapping $\sigma : x \mapsto \sigma \cdot x$ or $\sigma : y \mapsto x \cdot 1$ to any $\sigma \in S_n$ and any $x$ in $X$ (resp. any $y$ in $Y$). Let $f$ be a map from $X$ to $Y$. We say that $f$ is

- **$S_n$-invariant** if $\forall \sigma \in S_n, \forall x \in X, f(\sigma \cdot x) = f(x)$,
- **$S_n$-equivariant** if $\forall \sigma \in S_n, \forall x \in X, f(\sigma \cdot x) = \sigma f(x)$

Let us consider the natural permutation of nodes in the fixed layer. In this case the action is written explicitly as $\sigma : (x_i)_{i=1}^n \mapsto (x_{\sigma^{-1}(i)})_{i=1}^n$, for $x = (x_i)_{i=1}^n \in \mathbb{R}^n$. Like this action, if the action is caused by the permutation of the index set, i.e $\sigma : (x_i)_{i=1}^n \mapsto (x_{\sigma(i)})_{i=1}^n$ holds, we say that $\sigma$ is the $S_n$-action. We say that a deep neural network is $S_n$-equivariant (resp. $S_n$-invariant) if there are $S_n$-actions acting on each layer $\mathbb{R}^d$ and the corresponding map $Z_i : \mathbb{R}^d \rightarrow \mathbb{R}^{d+1}$ is $S_n$-equivariant (resp. $S_n$-invariant). We say that a deep neural network is $S_n$-equivariant if there are $S_n$-actions acting on each layer $\mathbb{R}^d$ and a natural number $c$ such that the corresponding map $Z_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_{i+1}}$ is $S_n$-equivariant for $1 \leq i \leq c$ and $S_n$-invariant for $c+1 \leq i \leq H$. We can easily confirm that the models in [Zaheer et al., 2018] satisfies these properties.

3 **Invariant case**

In this section, we discuss the invariant case. An important result about the structure of invariant functions is that they have an exact representation. Indeed, [Zaheer et al., 2018] have shown the following theorem:

**Theorem 3.1** ([Zaheer et al., 2018] Kolomogolov-Arnold representation theorem for permutation actions). Let $K \subset \mathbb{R}^n$ be a compact set. Then any continuous $S_n$-invariant function $f : K \rightarrow \mathbb{R}$ can be represented as

$$f(x_1, \ldots, x_n) = \rho \left( \sum_{i=1}^{n} \phi(x_i) \right)$$

for some continuous function $\rho : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$. Here, $\phi : \mathbb{R} \rightarrow \mathbb{R}^{n+1} : x \mapsto (1, x, x^2, \ldots, x^n)^\top$.

One of the important points of this theorem is that $\phi$ is independent of $f$. This representation theorem looks like a feed-forward network with two hidden layers, except that there is not a unique activation function. Then, the strategy of [Zaheer et al., 2018] to learn invariant functions is to replace $\rho$ and $\phi$ by universal approximators (in practice, a ReLU net with no more than 3 hidden layers) and then learn this approximators. Diagram 1 illustrates this idea.

**Theorem 3.2** (Permutation invariant version of universal approximation theorem). Let $K$ be a compact set in $\mathbb{R}^n$. Then for any $f : K \rightarrow \mathbb{R}^N$ which is continuous and $S_n$-invariant and for any $\epsilon > 0$, there is an $S_n$-invariant ReLU neural network $N$ such that its represented function $R_N$ satisfies $||f - R_N||_{\infty} \leq \epsilon$. Furthermore, we can take $N$ as:

- $N$ has two hidden layers and the width is not bounded.
- or

**Diagram 1**: A neural network approximating $S_n$-invariant function $f$. In blue : the inputs, in red : the output, in green : $\rho$ and $\phi$ who have to be learned.

- The width is of $N$ is bounded above by $n(n+2)$ and the depth is not bounded.

**Sketch of proof.** We first reduce the theorem to the case of $N = 1$. Since we consider the $L^\infty$-norm, if all components of $||f - R_N||_\infty$ is bounded by $\epsilon$, then $||f - R_N||_\infty \leq \epsilon$ holds. Hence, we may assume $N = 1$. Then by Theorem 3.1, we have $f(x_1, \ldots, x_n) = \rho(\sum_{i=1}^{n} \phi(x_i))$. We consider the diagram above. In the diagram, $\phi$ and $\Sigma$ are concrete maps. Hence, we can write by a feed forward network. We give $S_n$-actions on this network. We give the natural $S_n$-action on the input layer. Then, $S_n \Sigma$ is apparently an invariant function, it remains to approximate $\phi$ and $\rho$. Since $\phi$ depends on a single variable, we can extend $S_n$-action on arbitrary approximations. [Sonoda and Murata, 2017], we can approximate $\phi$ and $\rho$ by shallow networks. Hence, we have deep neural network $N$ which has two hidden layers and the width is not bounded. By [Hanin and Sellke, 2018], we can respectively approximate each of $\phi$ and $\rho$ by some neural networks whose width are bounded by $n + 2$ and the depth is not bounded. Hence, we have deep neural network $N$ whose width is bounded by $n(n+2)$ and the depth is not bounded.

1. 

4 **Relation between invariant and equivariant functions**

Here, we present a relation between $S_n$-equivariant map on $\mathbb{R}^n$ to $\mathbb{R}^n$ and invariant function on $\mathbb{R}^n$ to $\mathbb{R}$ by the action of the stabilizer subgroup $\text{Stab}_n(1)$ of $S_n$ with respect to $\{1\}$. Using this relation, we can reduce an approximation problem of $S_n$-equivariant maps to one of $\text{Stab}_n(1)$-invariant functions. In principle, this relation comes from the coset decomposition of $S_n$ by $\text{Stab}_n(1)$. We give a precise description of this relation below. For the details of permutation group $S_n$, see [Kurzweil and Stellmacher, 2006, Section 4.3].

The stabilizer subgroup of $S_n$ with respect to $\{1\}$ is defined by

$$\text{Stab}_n(1) = \{ \sigma \in S_n \mid \sigma \cdot 1 = 1 \}.$$  

When there is no ambiguity, we denote it by $\text{Stab}(1)$.
5 Equivariant case

In this section, we prove the equivariant version of the universal approximation theorem.

**Theorem 5.1** (Permutation equivariant version of universal approximation theorem). Let $K$ be a compact set in $\mathbb{R}^n$. Then for any $f : K \rightarrow \mathbb{R}^N$ which is continuous and $S_n\text{-equivariant}$ and for any $\epsilon > 0$, there is a $S_n\text{-equivariant}$ ReLU neural network $N$ such that its represented function $R_N$ satisfies $\|f - R_N\|_\infty \leq \epsilon$. Furthermore, we can take $N$ as:

- $N$ has two hidden layers and the width is not bounded.
- or
- The width of $N$ is bounded above by $n^3$ and the depth is not bounded.

More concretely, we construct a deep neural network made from stacking $S_n\text{-equivariant}$ single layers approximating a given $S_n\text{-equivariant}$ map. To achieve this, we divide the proof to five steps as follows:

1. By Proposition 4.1, we reduce the argument on $S_n\text{-equivariant}$ map $F$ to this of $\text{Stab}(1)$-invariant function $f$.
2. By a modification of Theorem 3.1, we have a representation of $\text{Stab}(1)$-invariant function $f$.
3. Using the above representation, we have a $\text{Stab}(1)$-invariant deep neural network which approximates $f$ and construct a deep neural network approximating $F$.
4. We introduce a certain action of $S_n$ on $(\mathbb{R}^n)^n$ which appears the first hidden layer naturally.
5. We show the $S_n\text{-equivariance}$ between the input layer and the first hidden layer.

Because Step 1 has already been explained in Section 4, we start from Step 2 in Section 5.1. Section 5.2, we explain Step 3. Section 5.3 is devoted to Step 4. In Section 5.4, we show Step 5 and conclude the proof of Theorem 5.1.

**5.1 Representation of $\text{Stab}(1)$-invariant functions**

We show a representation theorem of $\text{Stab}(1)$-invariant functions here. The stabilizer subgroup $\text{Stab}_n(1)$ is isomorphic to $S_{n-1}$ as a group. Hence, we can regard the $\text{Stab}(1)$-invariant function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ as an $S_{n-1}$-invariant function. This point of view allows us to use Theorem 3.1. Hence, we have the following representation theorem of $\text{Stab}(1)$-invariant functions as a corollary of Theorem 3.1.

**Corollary 5.1** (Representation of $\text{Stab}(1)$-invariant function). Let $K \subset \mathbb{R}^n$ be a compact set, let $f : K \rightarrow \mathbb{R}$ be a continuous and $\text{Stab}(1)$-invariant function. Then, $f(x)$ can be represented as

$$f(x_1, \ldots, x_n) = \rho \left( x_1, \sum_{i=2}^{n} \phi(x_i) \right),$$

for some continuous function $\rho : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$. Here, $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is similar as in Theorem 3.1.

By this corollary, we can represent any $\text{Stab}(1)$-invariant function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ as $f = \rho \circ L \circ \Phi$, where $\Phi : \mathbb{R} \rightarrow \mathbb{R} \times (\mathbb{R}^n)^{n-1}$ is

$$\Phi(x_1, \ldots, x_n) = (x_1, \phi(x_2), \ldots, \phi(x_n)),$$

and $L : \mathbb{R} \times (\mathbb{R}^n)^{n-1} \rightarrow \mathbb{R} \times \mathbb{R}^n$ is

$$L(x, (y_1, \ldots, y_{n-1})) = \left( x, \sum_{i=1}^{n-1} y_i \right).$$

**5.2 Universal approximation theorem for $\text{Stab}(1)$-invariant functions**

Here, we show that the existence of deep neural network approximating $f$. After that, using this approximator, we construct a deep neural network approximating $F$. By a slight modification of the invariant version of Theorem 3.2 for $\text{Stab}(1)$-invariant case, there exists sequence of deep neural network $\{A_m\}_m$ (resp. $\{B_m\}_m$) which converges to $\Phi$ (resp. $\rho$) uniformly. Then, the sequence of deep neural networks $\{B_m \circ L \circ A_m\}_m$ converges to $f = \rho \circ L \circ \Phi$ uniformly.

Now, $f$ can be approached by the following deep neural network by replacing $\rho$ and $\Phi$ by universal approximators as Diagram 2.

We remark that the left part (the part of before taking sum) of this deep neural network is naturally equivariant for the action of $\text{Stab}(1)$. For an $S_n\text{-equivariant}$ map $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$
with the natural action, by Proposition 4.1, there is a unique \( \text{Stab}(1) \)-invariant function \( f \) such that

\[
F(x) = \begin{pmatrix}
    f(x) \\
    f \circ (1 2)(x) \\
    f \circ (1 n)(x)
\end{pmatrix} = \begin{pmatrix}
    f \\
    (1 1) \circ f \\
    (1 2) \circ f
\end{pmatrix} \circ \begin{pmatrix}
    I_n \\
    I_n \\
    I_n
\end{pmatrix}(x).
\]

Here, we regard any element of \( S_n \) as a map from \( \mathbb{R}^n \) to \( \mathbb{R}^n \) and \( \circ \) means the composition of maps and \( I_n \) is the \( n \times n \) unit matrix. By the argument in this subsection, we can approximate \( f \) by the previous deep neural network \( \{B_m \circ L \circ A_m\}_m \). Substituting \( B_m \circ L \circ A_m \) for \( f \), we construct a deep neural network approximating \( F \) as Diagram 3.

The represented function of this neural network is

\[
\begin{pmatrix}
    B_m \circ L \circ A_m \\
    B_m \circ L \circ A_m \\
    B_m \circ L \circ A_m
\end{pmatrix} \circ \begin{pmatrix}
    (1 1) \\
    (1 2) \\
    (1 n)
\end{pmatrix} \circ \begin{pmatrix}
    I_n \\
    I_n \\
    I_n
\end{pmatrix}.
\]

The map \( F \) can be divide two parts, the part of transpositions and part of \( (f, f, \ldots, f) \). On the deep neural network (1) corresponding \( F \), the latter part corresponds to the layers from the first hidden layer to the output layer. Because this part is the \( n \) copies of same \( \text{Stab}(1) \)-invariant deep neural network (an approximation of \( (f, f, \ldots, f)^T \)) is clearly made of equivariant stacking layers for the natural action of \( S_n \). Therefore, it is remained to show that the former part is also \( S_n \)-equivariant.

We here investigate the width and the depth when we limit either. By Theorem 3.2, each of \( \phi \) and \( \rho \) can be approximated by a shallow neural network. Hence, if we do not bound the width, we can obtain deep neural network approximating \( F \) with depth 3. On the other hand, by Theorem 3.2 again, if we do not bound the depth, \( \phi \) (resp. \( \rho \)) can be approximated by a deep neural network with width \( n + 1 \) (resp. \( n + 2 \)). Thus, we can obtain a deep neural network approximating \( F \) with width bounded above by \( n^3 \).

5.3 Another \( S_n \)-action *

As mentioned above, we show that our deep neural network is actually made of stacking \( S_n \)-equivariant layers. The most difficult part is to show the equivariance between the input layer and the first hidden layer which induces a function \( g \) from \( \mathbb{R}^n \) to a certain direct sum of copies of the same spaces \( V^n \). However, the natural action on that last space doesn’t make the layer being equivariant. For this reason, we need to define a new action of \( S_n \) on \( V^n \) exploiting the \( \text{Stab}(1) \)-equivariance among each copies.

**Definition 5.1.** Let \( V \) be \( n \)-dimensional real vector space. We suppose that \( \text{Stab}(1) \) acts on each \( V \), denoted as \( \sigma \cdot x \). Then, we define an action of \( S_n \) on \( V^n \) as follows:

\[
\sigma \star \begin{pmatrix}
    x_1 \\
    \vdots \\
    x_n
\end{pmatrix} = \begin{pmatrix}
    \tilde{\sigma}_1 \cdot x_{\sigma^{-1}(1)} \\
    \vdots \\
    \tilde{\sigma}_n \cdot x_{\sigma^{-1}(n)}
\end{pmatrix}
\]

for any \( \sigma \in S_n \), and for any \( (x_1, x_2, \ldots, x_n) \in V^n \). Here, for any \( i \), \( \tilde{\sigma}_i \) is an element of \( \text{Stab}(1) \) defined as \( (1 i)\sigma = \tilde{\sigma}_i (1 \sigma^{-1}(i)) \).

We postpone the proof of well-definedness of this left action “\( \star \)” to Appendix C. We should remark that this action “\( \star \)” naturally appears in representation theory as the induced representation, which is an operation to construct a representation of group \( G \) from a representation of a subgroup \( H \) of \( G \). In this case, the action “\( \star \)” is the induced representation of \( \text{Stab}(1) \) defined by the action “\( \cdot \).”

5.4 Equivariance between the input layer and the first hidden layer

We conclude the proof of Theorem 5.1 by checking the remained part, the equivariance between the input layer \( \mathbb{R}^n \) and the first hidden layer \( V^n \) of our deep neural network (1). (Although \( V \) is equal to \( \mathbb{R}^n \), we distinguish them to stress the difference.) The situation is as follows: \( S_n \) acts on \( \mathbb{R}^n \) by the natural action “\( \cdot \)”, and on \( V^n \) by the action “\( \star \)” in Definition 5.1. We remark that \( \text{Stab}(1) \) acts on \( V \) by “\( \cdot \)” in this setting, it suffices to show the \( S_n \)-equivariance of the following map \( g \): For a \( \text{Stab}(1) \)-equivariant linear function \( l : \mathbb{R}^n \rightarrow V \),
the map \( g : \mathbb{R}^n \to V^n \) is defined by
\[
g = \begin{pmatrix} \text{ReLU} \\ \text{ReLU} \end{pmatrix} \circ \begin{pmatrix} I \\ l \\ \vdots \\ \vdots \\ I \\ l \\ \vdots \\ \vdots \\ I \end{pmatrix} \circ \begin{pmatrix} I_n \\ \vdots \\ \vdots \\ I_n \end{pmatrix}.
\]

For any \( i \), any \( \sigma \in S_n \) and any \( x \in \mathbb{R}^n \), we have
\[
(l \circ (1i) \circ I_n)(\sigma \cdot x) = l(((1i)\sigma) \cdot x).
\]

Because for any \( i \), there is a unique \( \tilde{\sigma}_i \in \text{Stab}(1) \) such that \((1i)\sigma = \tilde{\sigma}_i(1\sigma^{-1}(i))\) as in Definition 5.1, we have
\[
l(((1i)\sigma) \cdot x) = l(\tilde{\sigma}_i(1\sigma^{-1}(i)) \cdot x) = \tilde{\sigma}_i \cdot l((1\sigma^{-1}(i)) \cdot x).
\]
The last equality is due to \( \text{Stab}(1) \)-equivariance of \( l \). On the other hand, by Definition 5.1, \( i \)-th entry of \( \sigma \cdot g(x) \) becomes
\[
(\sigma \cdot g(x))_i = \tilde{\sigma}_i \cdot \text{ReLU}(l((1\sigma^{-1}(i)) \cdot x)).
\]
Because \( \tilde{\sigma}_i \circ \text{ReLU} = \text{ReLU} \circ \tilde{\sigma}_i \) holds, \( g \) is \( S_n \)-equivariant.
We conclude the proof of Theorem 5.1.

6 Dimension reduction and data augmentation

In this section, we discuss about the advantages of the invariant/equivariant models. We first point out that the invariant/equivariant models have lower parameters than the usual models if the architecture as a deep neural network is same. This is because they have the constraint conditions induced by the invariance/equivariance of permutations. Indeed, the following proposition calculates the number of parameters in the equivariant models.

**Proposition 6.1.** Let \( Z_i : \mathbb{R}^M \to \mathbb{R}^N \) be the map appeared in a layer in the equivariant models. Then \( n \) devides \( M \) and \( N \), and the number of the free parameters in \( W_i \) is equal to \( 2M N/n^2 \).

**Sketch of proof.** Since \( \mathbb{R}^M \) and \( \mathbb{R}^N \) have \( S_n \)-action, by considering the orbit of the coordinates, we see that \( n \) devides \( M \) and \( N \). Let us write \( \mathbb{R}^M = (\mathbb{R}^n)^M \) and \( \mathbb{R}^N = (\mathbb{R}^n)^N \). In this case, \( W_i \) is written by sum of \( n \times n \) matrices and each \( n \times n \) matrix \( V_i \) corresponds to the linear map: \( \mathbb{R}^n \to (\mathbb{R}^n)^M, Z \to (\mathbb{R}^n)^N \to \mathbb{R}^n \), where the first map is the inclusion to coordinates of \( (\mathbb{R}^n)^M \) and the last map is projection to coordinates of \( (\mathbb{R}^n)^N \). Since these constructions are taken to be compatible with \( S_n \)-action, we see that \( \text{ReLU} \circ V_i \) is \( S_n \)-equivariant. If the activation functions are bijective, we are done because of the same discussion as in the proof of Lemma 3 in [Zaheer et al., 2018]. But since \( \text{ReLU} \) functions are not bijective, we need more discussion. Let us take a transposition \((p,q)\). We consider the condition under which the \( p \)-th coordinate of \( V_i(x) \) is positive. Since the \( p \)-th coordinate of \( V_i(x) \) is written as an inner product of the \( p \)-th row vector of \( V_i \) and \( x \), the points in the upper half of the hyperplane defined by \( p \)-th row vector give positive values. Hence, if \( x \) is in the intersection of two hyperplane associated to \( p \) and \( q \), the \( p \)-th coordinate and \( q \)-th coordinate of \( V_i(x) \) is positive. This means that if \( x \) is in the intersection of two hyperplane and if we consider the transition \((p,q)\), we may assume \( V_i = W_i \). Combining with the discussion in the proof of Lemma 3 in [Zaheer et al., 2018], we have \( V_i = M + \gamma(1^1) \). Hence, for each \( n \times n \) matrix, we have two free parameters. Since the number of \( n \times n \) matrices appeared in \( W_i \) is \( MN/n^2 \), we have the desired result.

By this observation, we conclude the number of free parameters in the equivariant models.

**Theorem 6.1.** Let \( N \) be a deep neural network of the invariant model, consist of the equivariant part of depth \( d \) and width \( M \) and the preinvariant part of depth \( e \) and width \( N \) (resp. the equivariant model of depth \( d \) and width \( M \)). Then, the number of free parameters in the model is \( (2^d M^2 d/n^2)^e \cdot N^{2e} \) (resp. \( 2^d M^{2d}/n^{2d} \)).

The number of the free parameters in the usual model is \( M^{2d} \). Hence, this theorem implies that the free parameters of the invariant/equivariant models are exponentially fewer than the one of the usual models.

Data augmentation is a common technique in empirical learning. In the case of the invariant/equivariant tasks, a possible augmentation is to make new samples by the acting permutation on samples. New samples are effective to the usual models, but not effective to our invariant/equivariant models. This is because in our models, all weights are symmetric under permutation actions. This means that our models learn augmented samples from a sample. By acting permutations, we can make \( n! \) new samples from a sample. Hence, computational complexity is reduced to \( 1/n! \) times.

7 Conclusion

We introduced the invariant/equivariant models. These models are the universal approximator for invariant/equivariant functions. These theorems and the discussion in section 6 give a guarantee for the success of the models in [Zaheer et al., 2018], i.e. although the free parameters of the invariant/equivariant models are exponentially fewer than the one of the usual models, the invariant/equivariant models can approximate the invariant/equivariant functions to arbitrary accuracy. Our theory also implies that there is much possibility that the group models behave as the usual models for the tasks related to groups. This must be a good perspective to develop the models in deep learning.

**References**

[Alex et al., 2012] Krizhevsky Alex, Sutskever Sutskever, Ilya, and Hinton Geoffrey E. Imagenet classification with convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[Arnold, 1957] V. I. Arnold. On functions of three variables. Proceedings of the USSR Academy of Sciences, 114:679–681, 1957. English translation: Amer. Math. Soc. Transl., 28 (1963), pp. 5154.

[Barron, 1994] Andrew R Barron. Approximation and estimation bounds for artificial neural networks. Machine learning, 14(1):115–133, 1994.
[Cohen and Welling, 2016] Taco S Cohen and Max Welling. Group equivariant convolutional networks. *Proceedings of the 33rd International Conference on Machine Learning*, 4, 2016. URL arXiv:arXiv:1602.07576, 2016.

[Cohen et al., 2016] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. *Conference on Learning Theory*, page 698728, 2016.

[Cybenko, 1989] G. Cybenko. Approximation by superposition of a sigmoidal function. 1989. Math. Control Signals systems (1989) 2: 303–314.

[Delalleau and Bengio, 2011] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. *Advances in Neural Information Processing Systems*, page 666674, 2011.

[Eldan and Shamir, 2016] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. *Conference on Learning Theory*, page 907940, 2016.

[Funahashi, 1989] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.

[Gens and Domingos, 2014] R. Gens and P Domingos. Deep symmetry networks. *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[Goodfellow et al., 2013] Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

[Hanin and Sellke, 2018] Boris Hanin and Mark Sellke. Approximating continuous functions by relu nets of minimal width. 2018. URL: https://arxiv.org/abs/1710.11278.

[Hornik et al., 1989] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[Kolmogorov, 1956] A. N. Kolmogorov. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. *Proceeding of the USSR Academy of Sciences*, 108:176–182, 1956. English translation: Amer. Math. Soc. Transl., 17 (1961), pp. 369373.

[Kurkova, 2002] Věra Kurková and Marcello Sanguineti. Comparison of worst case errors in linear and neural network approximation. *IEEE Transactions on Information Theory*, 48(1):264–275, 2002.

[Kurkova, 1992] V. Kurkova. Kolmogorovs theorem and multilayer neural networks. *Neural networks*, 5(3):501506, 1992.

[Kurzweil and Stellmacher, 2006] Hans Kurzweil and Bernd Stellmacher. *The theory of finite groups: an introduction*. Springer Science & Business Media, 2006.

[Lu et al., 2017] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. *Neural Information Processing Systems*, pages 6231–6239, 2017.

[Silver et al., 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[Sonoda and Murata, 2017] Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[Wan et al., 2013] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[Zaheer et al., 2018] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. 2018. URL: https://arxiv.org/abs/1703.06114.
A Proof of Proposition 3.2

Proof of Proposition 3.2. For \( N = 1 \), we have \( f : K \mapsto \mathbb{R} \), then, by Theorem 3.1, we obtain the representation \( f(x_1, \ldots, x_n) = \rho(\sum_{i=1}^{n} \phi(x_i)) \). By the universal approximation property of ReLU deep neural network, we can find two sequences of ReLU DNN \( \{N^F_\phi\}_k \) and \( \{N^\rho_\phi\}_k \) such that their corresponding functions \( \{F^F_\phi\}_k \) and \( \{F^\rho_\phi\}_k \) tend to \( \rho \) and \( \phi \) for the \( L^\infty \)-norm when \( k \) tends to infinity. Let \( \{N_k\}_k \) be the sequence of networks whose corresponding functions are \( F^F_\phi : (x_1, \ldots, x_n) \mapsto (\sum_{i=1}^{n} F^F_\phi(x_i)) \). To show that \( \{F^F_\phi\}_k \) uniformly tends to \( f \), we use the following lemma:

Lemma A.1. If \( \{f_k\}_k \) tends uniformly to \( f \), \( \{g_k\}_k \) tends uniformly to \( g \) and \( f \) is uniformly continuous, then \( \{f_k \circ g_k\}_k \) tends uniformly to \( f \circ g \).

Proof of Lemma A.1. For any \( k \), we have

\[
|f_k \circ g_k(x) - f \circ g(x)| = |f_k \circ g_k(x) - f \circ g_k(x) + f \circ g_k(x) - f \circ g(x)|
\leq |f_k \circ g_k(x) - f \circ g_k(x)| + |f \circ g_k(x) - f \circ g(x)|.
\]

Let \( \epsilon > 0 \). By the uniform continuity of \( f \), there is a \( \delta > 0 \) such that for all \( x, y \) satisfying \( |x - y| \leq \delta \), \( |f(x) - f(y)| \leq \epsilon/2 \) holds. Then for large enough \( k \), we have both \( ||g_k - g||_\infty \leq \delta \) which implies that for any \( x \), \( |f \circ g_k(x) - f \circ g(x)| \leq \epsilon/2 \), and \( ||f_k - f||_\infty \leq \epsilon/2 \). Hence, for any \( k \) large enough, \( ||f_k \circ g_k - f \circ g|| \leq \epsilon \) holds.

Now using Lemma A.1, we have that \( (x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n} F^F_\phi(x_i) \) tends to \( (x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n} \phi(x_i) \), because \( (x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n} x_i \) is Lipschitz (by triangular inequality) so uniformly continuous. Then, using Lemma A.1 again, we obtain the result, since \( \rho \) is continuous on a compact set so uniformly continuous. Moreover, by [Sonoda and Murata, 2017], we can approximate \( \phi \) and \( \rho \) by shallow networks. Hence, we have an approximation by a deep neural network \( \mathcal{N} \) which has two hidden layers and the width is not bounded. By [Hanin and Sellke, 2018], we can respectively approximate each of \( \phi \) and \( \rho \) by some neural networks whose width are bounded by \( n + 2 \) and the depth is not bounded. Hence, we have an approximation by a deep neural network \( \mathcal{N} \) whose width is bounded by \( n(n + 2) \) and the depth is not bounded.

The invariant function \( f \) is approached by a deep neural network \( \mathcal{N} \) having the following diagram:

\[
\begin{array}{c}
\text{\(x_1\)} \quad \mathcal{N}^1 \quad \mathcal{N}^2 \quad \mathcal{N}^3 \\
\text{\(x_2\)} \quad \mathcal{N}^1 \quad \mathcal{N}^2 \quad \mathcal{N}^3 \\
\vdots \quad \mathcal{N}^1 \quad \mathcal{N}^2 \quad \mathcal{N}^3 \\
\vdots \quad \mathcal{N}^1 \quad \mathcal{N}^2 \quad \mathcal{N}^3 \\
\text{\(x_n\)} \quad \mathcal{N}^1 \quad \mathcal{N}^2 \quad \mathcal{N}^3 \\
\end{array}
\]

Let us show that this is a \( S_n \)-invariant deep neural network. Since the sum \((\Sigma)\) is an invariant function, we can divide this neural network in two parts: the fist part on the left of the symbol \( \Sigma \) and the second on the right. For each layer \( \mathbb{R}^{d_i} \) of \( \mathcal{N}^1 \), there is a corresponding map \( Z_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_{i+1}} \). Then for each layer \((\mathbb{R}^{d_i})^n \) of the first part of \( \mathcal{N} \), there is a \( S_n \) action \( \sigma \cdot (x_i)_i = (x_{\sigma^{-1}(i)})_i \) for \( x = (x_i)_i \in \mathbb{R}^{d_i} \), and the corresponding map \((Z_1, \ldots, Z_n) : (\mathbb{R}^{d_i})^n \rightarrow (\mathbb{R}^{d_{i+1}})^n \) is \( S_n \)-equivariant. On the second part of \( \mathcal{N} \), there is no \( S_n \) actions on the layers \( \mathbb{R}^{d_i} \) except the trivial action. Hence, for this action, each corresponding map \( Z_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_{i+1}} \) is invariant. This shows that the network \( \mathcal{N} \) is \( S_n \)-invariant.

B Proof of Proposition 4.1

Proof of Proposition 4.1. First, we show that for \( \text{Stab}(1) \)-invariant function \( f \), the map \( F = (f \circ (1 i))_i \in \mathbb{R}^n \) is \( S_n \)-equivariant. It is well-known that the transpositions \((i j)\) for \( i, j = 1, \ldots, n \) generate \( S_n \). Any transposition \((i j)\) can be written as \((i j) = (1 i)(1 j)(1 i)\), hence \((1 i) = 1, \ldots, n \) also generate \( S_n \). Therefore, it suffices to show that \( F((1 i) \cdot x) = (1 i) \cdot F(x) \) for any \( i \) and any \( x \in \mathbb{R}^n \). If \( j \) is not equal to \( i \), we have

\[
F((1 i) \cdot x)_j = f((1 j)((1 i) \cdot x))
\]
Here, $F(x)_j$ means the $j$-th entry of $F(x) \in \mathbb{R}^n$. The last equation is due to Stab(1)-invariance of $f$. Similarly, we have $F((1 i) \cdot x)_i = f(x) = F(x)_1$ and $F((1 i) \cdot x)_1 = f((1 i) \cdot x) = F(x)_i$. Hence $F((1 i) \cdot x) = (1 i) \cdot F(x)$ holds.

Next, we assume that $F$ is $S_n$-equivariant, i.e. $F(\sigma \cdot x) = \sigma \cdot F(x)$ for any $x \in \mathbb{R}^n$ and any $\sigma \in S_n$. We set $F = (f_1, \ldots, f_n)^T, f_i : \mathbb{R}^n \to \mathbb{R}$. Then, by the definition, the $S_n$-equivariance of $F$ is equivalent to

$$f_i(\sigma \cdot x) = f_{\sigma^{-1}(i)}(x) \tag{2}$$

for any $\sigma \in S_n$ and $x \in \mathbb{R}^n$. By this equation (2) with $i = 1$, we have

$$f_1(\sigma \cdot x) = f_{\sigma^{-1}(1)}(x)$$

for any $\sigma \in S_n$ and any $x \in \mathbb{R}^n$. Because $\sigma \in \text{Stab}(1)$ if and only if $\sigma^{-1} \in \text{Stab}(1)$, we have $f_{\sigma^{-1}(1)}(x) = f_1(x)$ if $\sigma \in \text{Stab}(1)$. Thus, $f_1$ is Stab(1)-invariant. We consider the equation (2) for $i = 1$ and the transposition $\sigma = (1 j)$ for $j \neq 1$, we have

$$f_1((1 j) \cdot x) = f_1((1 j)^{-1}(1)) = f_j(x).$$

This means $F(x) = (f_1((1 i) \cdot x))_i$. The unicity of $f = f_1$ for $F$ is trivial by the above argument. □

C Well-definedness of the action “*”.

We here show that the left action “*” of $S_n$ on $V^n$ defined in Section 5.3 is well-defined, i.e. for any $\sigma, \tau \in S_n$ and any $X = (x_1^T, \ldots, x_n^T)^T \in V^n$, we have $\tau * (\sigma * X) = (\tau \sigma) * X$.

The permutation group $S_n$ is decomposed as

$$S_n = \bigcup_{i=1}^n \text{Stab}(1)(1 i).$$

For any $\sigma \in S_n$, because $\sigma(1 \sigma^{-1}(1))$ is in Stab(1), this $\sigma$ is in the coset $\text{Stab}(1)(1 \sigma^{-1}(1))$. Apply this for $(1 i)\sigma$ for $i$, $(1 i)\sigma$ is in the coset

$$\text{Stab}(1)(1 ((1 i)\sigma)^{-1}(1)) = \text{Stab}(1)(1 \sigma^{-1}(i)),$$

thus, we have a unique element $\tilde{\sigma}_i \in \text{Stab}(1)$ such that

$$\tilde{\sigma}_i = (1 i)\sigma(1 \sigma^{-1}(i)). \tag{3}$$

For $\sigma, \tau \in S_n$ and $X = (x_1^T, \ldots, x_n^T)^T \in V^n$, we have

$$\tau * (\sigma * X) = \tau * (\tilde{\sigma}_1 \cdot x_{\sigma^{-1}(1)} \cdot \tilde{\sigma}_2 \cdot x_{\sigma^{-1}(2)} \cdot \cdots \cdot \tilde{\sigma}_n \cdot x_{\sigma^{-1}(n)}) = \tau * (\tilde{\sigma}_1 \cdot x_{\tau^{-1}(1)} \cdot \tilde{\sigma}_2 \cdot x_{\tau^{-1}(2)} \cdot \cdots \cdot \tilde{\sigma}_n \cdot x_{\tau^{-1}(n)}).$$

Then, by the equation (3), $\tilde{\sigma}_i = (1 i)\sigma(1 \sigma^{-1}(i))$. Hence we have

$$\tilde{\sigma}_i \tau^{-1}(i) = (1 i)\tau(1 \tau^{-1}(i))(1 \tau^{-1}(i))\sigma(1 \sigma^{-1}(\tau^{-1}(i))) = (1 i)\tau\sigma(1 \tau^{-1}(i)) = \tilde{\sigma}_i.$$

This relation implies

$$\tau * (\sigma * X) = (\tilde{\sigma}_1 \cdot x_{\tau^{-1}(1)} \cdot \tilde{\sigma}_2 \cdot x_{\tau^{-1}(2)} \cdot \cdots \cdot \tilde{\sigma}_n \cdot x_{\tau^{-1}(n)}) = (\tilde{\sigma}_1 \cdot x_{\tau^{-1}(1)} \cdot \tilde{\sigma}_2 \cdot x_{\tau^{-1}(2)} \cdot \cdots \cdot \tilde{\sigma}_n \cdot x_{\tau^{-1}(n)}) = (\tau \sigma) * X.$$

Thus, the action is well-defined.
D Dimension reduction

In this section, we give the proof of Proposition 6.1.

Proof of Proposition 6.1. We firstly see that $n$ divides $M$ and $N$. By induction on $L$, we claim that if $\mathbb{R}^L$ has $S_n$-action, $n$ divides $L$. If $L$ is smaller than $n$, it is easy to see that $\mathbb{R}^L$ does not have $S_n$-action. Let $(x_i)_{i \in I}$ be the coordinate of $\mathbb{R}^L$. Since $\mathbb{R}^L$ have $S_n$-action, for each $\sigma \in S_n$, there is a $j_\sigma$ such that $\sigma \circ x_1 = x_{j_\sigma}$ holds. We can see that the set $J_1 = \{ j_\sigma | \sigma \in S_n \}$ is consist of $n$ elements. Let $J_1 = I - J_1$, then $(x_i)_{i \in I_1}$ is the coordinate of $\mathbb{R}^{L-n}$ and $\mathbb{R}^{L-n}$ has $S_n$-action induced by the one of $\mathbb{R}^L$. By the inductive hypothesis, $n$ divides $L - n$, hence $n$ divides $L$.

Let us write $\mathbb{R}^M = (\mathbb{R}^n)^{M'}$ and $\mathbb{R}^N = (\mathbb{R}^n)^{N'}$. In this case, $W_i$ can be written as the following form:

\[
W_i = \begin{pmatrix}
V_{i1} & V_{i2} & \cdots & V_{iM'} \\
V_{i1} & V_{i2} & \cdots & V_{i2M'} \\
\vdots & \vdots & \ddots & \vdots \\
V_{iM'} & V_{iM'+1} & \cdots & V_{iN'M'}
\end{pmatrix}
\]

where $V_{ij}$ are $n \times n$ matrices. Let us consider the following maps:

\[
\mathbb{R}^n \xrightarrow{i} (\mathbb{R}^n)^{M'} \xrightarrow{Z_i} (\mathbb{R}^n)^{N'} \xrightarrow{p} \mathbb{R}^n
\]

where the first map is the inclusion to the coordinates started from the $(j-1)n$ th coordinate of $(\mathbb{R}^n)^{M'}$ ended at the $jn-1$ th coordinate of $(\mathbb{R}^n)^{M'}$ and the last map is the projection to the coordinates started from the $(i-1)n$ th coordinate of $(\mathbb{R}^n)^{N'}$ ended at the $in-1$ th coordinate of $(\mathbb{R}^n)^{N'}$. We chase the elements of these maps as follows;

\[
p \circ Z_i \circ i \left( \begin{array}{c}
a_1 \\
a_2 \\
\vdots \\
a_n
\end{array} \right) = p \circ Z_i \left( \begin{array}{c}
0 \\
\vdots \\
a_1 \\
\vdots \\
a_n \\
0
\end{array} \right)
\]

\[
= p \begin{pmatrix}
\text{ReLU} \circ \begin{pmatrix}
V_{i1} & V_{i2} & \cdots & V_{iM'} \\
V_{i1} & V_{i2} & \cdots & V_{i2M'} \\
\vdots & \vdots & \ddots & \vdots \\
V_{iM'} & V_{iM'+1} & \cdots & V_{iN'M'}
\end{pmatrix}
\end{pmatrix}
\begin{pmatrix}
0 \\
\vdots \\
a_1 \\
\vdots \\
a_n \\
0
\end{pmatrix}
\]

\[
= p \begin{pmatrix}
\text{ReLU} \circ V_{ijj} \\
\vdots \\
\text{ReLU} \circ V_{ijp}
\end{pmatrix}
\begin{pmatrix}
a_1 \\
\vdots \\
a_n
\end{pmatrix}
\]

\[
= \text{ReLU} \circ V_{ijj} \\
\vdots \\
\text{ReLU} \circ V_{ijp}
\begin{pmatrix}
a_1 \\
\vdots \\
a_n
\end{pmatrix}
\]
Hence, each $n \times n$ matrix $V_{ij}$ induces a subneural network. Since these constructions are taken to be compatible with $S_n$-action, we see that $f = \text{ReLU} \circ V_{ij} : \mathbb{R}^k \to \mathbb{R}^k$ is $S_n$-equivariant. If the activation functions are bijective, we are done because of the same discussion as in the proof of Lemma 3 in [Zaheer et al., 2018]. But since ReLU functions are not bijective, we need more discussion. Let us take a transposition $\sigma = (p, q)$. Since $f$ is $S_n$-equivariant, $\sigma \circ f(x) = f(\sigma \circ x)$ holds for any $x$. We have

$$
\sigma \circ f(x) = \sigma \left( \text{ReLU} \circ \begin{pmatrix}
    a_{11} & \cdots & a_{1n} \\
    \vdots & \ddots & \vdots \\
    a_{p1} & \cdots & a_{pn} \\
    \vdots & \ddots & \vdots \\
    a_{q1} & \cdots & a_{qn} \\
    \vdots & \ddots & \vdots \\
    a_{n1} & \cdots & a_{nn}
\end{pmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}
\right)
$$
We have that the coefficients of each equation coincide. Hence, we have the desired result. This implies that the zero set of the equations are the same. But it is clear that the equations are positive for some \(x\). Hence, we can write

\[
\begin{pmatrix}
a_{11} & \cdots & a_{1q} & \cdots & a_{1p} & \cdots & a_{1n}
\end{pmatrix}
\begin{pmatrix}
x_1
\end{pmatrix}
= \text{ReLU}
\begin{pmatrix}
a_{11} & \cdots & a_{1p} & \cdots & a_{1q} & \cdots & a_{1n}
\end{pmatrix}
\begin{pmatrix}
x_1
\end{pmatrix}
\begin{pmatrix}
a_{21} & \cdots & a_{2q} & \cdots & a_{2p} & \cdots & a_{2n}
\end{pmatrix}
\begin{pmatrix}
x_2
\end{pmatrix}
= \text{ReLU}
\begin{pmatrix}
a_{21} & \cdots & a_{2p} & \cdots & a_{2q} & \cdots & a_{2n}
\end{pmatrix}
\begin{pmatrix}
x_2
\end{pmatrix}
\begin{pmatrix}
a_{n1} & \cdots & a_{nq} & \cdots & a_{np} & \cdots & a_{nn}
\end{pmatrix}
\begin{pmatrix}
x_n
\end{pmatrix}
= \text{ReLU}
\begin{pmatrix}
a_{n1} & \cdots & a_{nq} & \cdots & a_{np} & \cdots & a_{nn}
\end{pmatrix}
\begin{pmatrix}
x_n
\end{pmatrix}
\]

for any \(x\). Hence, the \(l\)th coordinate of the left hand side is positive if and only if the one of the right hand side is positive. This implies that the zero set of the equations are the same. But it is clear that the equations are positive for some \(x\). This implies that the coefficients of each equation coincide. Hence, we have the desired result.

We show that \(a_{pp} = a_{qq}\) holds for any \(p, q\). We can see that the \((p, q)\) entry of the matrix of the left hand side in the claim is equal to \(a_{pp}\). Similarly, the \((p, q)\) entry of the matrix of the right hand side in the claim is equal to \(a_{qq}\). Hence, by the claim, we have \(a_{pp} = a_{qq}\). We show that if \(i \neq j\) and \(s \neq t\), \(a_{ij} = a_{st}\) holds. Consider the \((i, q)\) entry of each matrix, the one of the left hand side is equal to \(a_{ip}\) and the one of the right hand side is equal to \(a_{iq}\). Hence, we have \(a_{ip} = a_{iq}\), where \(i \neq p\) and \(i \neq q\). By the symmetry, \(a_{pi} = a_{qi}\) holds for any \(i \neq p\) and \(i \neq q\). Hence, we have

\[a_{ij} = a_{ij} = a_{st}\]

for any \(i \neq j\) and \(s \neq t\). Hence, we can write \(V_{ij} = \lambda I + \gamma (11^T)\). 

\[\square\]