# Group Theory in Machine Learning

Atharv Chagi

## Introduction

As of recent years, there has been no question of the dominance of machine learning and artificial intelligence. With significant advances in computer vision, natural language processing, and robotics, the field has been growing at a rapid rate. However, many machine learning models have a reliance on large datasets and computational power. To handle data transformations like rotations, translations, and reflections requires extensive data augmentation.

Exploiting symmetries is rooted in group theory and can be used to encode invariances and equivariances into machine learning models. This reduces redundant computations and increases the interpretability of the training algorithms.

In real world data, symmetries are everywhere. For example, when you take an image and rotate it, it is still the same image. Traditionally to address this, data will be augmented with all possible permutations of the image. If done for every image in a large dataset, it becomes very computationally expensive to train.

Assuming core machine learning ideas, this paper aims to talk about the utility of group theory in machine learning, focusing on how the mathematical principles of symmetry and transformations improve machine learning models. It will speak about group theory and representations then highlight key applications in machine learning.

## Group Theory and Representations

We can begin by describing a group. A **group** is a set $G$ equipped with a binary operation that satisfies the following properties:

- Closure: For all $a, b \in G$, $a \cdot b \in G$
- Associativity: For all $a, b, c \in G$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Identity: There exists an element $e \in G$ such that for all $a \in G$, $e \cdot a = a \cdot e = a$
- Inverses: For all $a \in G$, there exists an element $b \in G$ such that $a \cdot b = b \cdot a = e$

In machine learning, groups are often used to describe symmetries in data. For example, if we have a dataset of images of a face, we can describe the symmetries of the face using a group. We could use the set of rotations SO(2) to describe the symmetries of the face. These 2x2 rotation matrices of the form:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

It is also important to speak about representation theory. **Representation theory** studies how groups can act on vector spaces. Essentially, every element of a group is mapped to a matrix.

More formally, a **representation** of a group $G$ is a homomorphism $\rho : G \to GL(V)$ where $V$ is a vector space and $GL(V)$ is the group of invertible linear transformations on $V$. Every group element $g \in G$ is mapped to a matrix $\rho(g) \in GL(V)$ that acts on the vectors $v \in V$. This is a **matrix representation**.

$$\rho(g) : V \to V \quad v \mapsto \rho(g)v$$

There are also building blocks to these representations called **irreducible representations**. We say a representation $\rho$ is irreducible if the only subspaces of $V$ that are invariant under the action of $\rho(G)$ are the trivial subspaces $\{0\}$ and $V$ itself.

It is important to note that not all representations are irreducible; in fact, most representations are reducible. That being said, any representation can be decomposed into a direct sum of irreducible representations.

$$\rho = \bigoplus_i \rho_i$$

As an example, consider $C_n$, the cyclic group of order $n$. $C_n = \{e, g, g^2, ..., g^{n-1}\}$ where $g^n = e$. Here, the irreducible representations are just the rotations of the $n$-gon.

$$\rho_k(g^j) = e^{2\pi ikj/n}$$

Each of these representations $\rho_k$ is irreducible since they map to one-dimensional complex numbers, and this representation is actually the foundation of the discrete Fourier Transform.

## Encoding Invariances

Representation theory provides a sublime way to encode symmetries into machine learning models. In many machine learning tasks, it is crucial to extract features from data that remain invaraint under certain transformations. For example, in image tasks, images should remain the same regardless of rotations and translations. Or in tasks with graphs, properties of the graph should remain the same if the nodes on a graph are permuted.

We can construct a feature transform on the data that gives data points which are invariant to a group's actions. For a group $G$, we can construct a feature transform $\phi : V \to V$ defined by:

$$\phi(v) = \frac{1}{|G|} \sum_{g \in G} \rho(g)v$$

This averaging over group actions ensures that the now engineered datapoints $\phi(v)$ are invariant to actions from the group.

### Encoding Equivariances

In some other applications, it could be desired to retain information about the transformation when processing data. **Equivariance** ensures that if an input data point $v$ is acted on by $g \in G$, then the extracted features $\psi(v)$ are transformed predictably.

$$\psi(\rho(g)v) = \rho'(g)\psi(v)$$

Where $\rho(g)$ and $\rho'(g)$ are representations of $G$ acting on the input and output spaces respectively.

## Applications in Machine Learning

A prominent example of equivariance in practice is AlphaFold 2, Google DeepMind's protein prediction model. As proteins are in 3D space, AlphaFold uses equivariant neural networks to make sure its predictions are consistent regardless of how the input protein sequence is permuted; in this case E(3)-equivariance. Equivariance is crucial for accurately predicting protein folding, as the physical properties and structure of a protein should not depend on its orientation in space.

Another example of equivariance in machine learning is Graph Neural Networks (GNNs). GNNs are designed to be equivariant to permutations of nodes in a graph. This means that if the nodes in the input graph are permuted, the output of the GNN will be permuted in exactly the same way. Mathematically, if $P$ is a permutation matrix and $f$ is a GNN:

$$f(PXP^T) = Pf(X)P^T$$

Where $X$ is the input graph's adjacency matrix. This equivariance property is crucial because the meaning of a graph should not depend on how the nodes are numbered. For example, in molecular property prediction, the properties of a molecule should not depend on how we order its atoms in our representation.

GNNs achieve this equivariance through message passing operations that only depend on the graph structure. A typical message passing layer looks like:

$$h_i^{(l+1)} = \phi\left(h_i^{(l)}, \sum_{j \in \mathcal{N}(i)} \psi(h_i^{(l)}, h_j^{(l)}, e_{ij})\right)$$

Where $h_i^{(l)}$ is the feature vector of node $i$ at layer $l$, $\mathcal{N}(i)$ are the neighbors of node $i$, $e_{ij}$ is the edge feature between nodes $i$ and $j$, and $\phi$ and $\psi$ are learnable functions. This structure naturally preserves the permutation equivariance of the network.

Image processing is another domain where equivariance plays a crucial role. Convolutional Neural Networks (CNNs) are naturally translation equivariant which makes CNNs powerful for image tasks. However, standard CNNs are not equivariant to other transformations like rotations or reflections. This limitation led to the development of Group Equivariant CNNs (G-CNNs) that are equivariant to larger groups of transformations. For instance, a rotation-equivariant CNN would recognize a rotated cat image just as well as the original, making the network more robust to variations in object orientation.

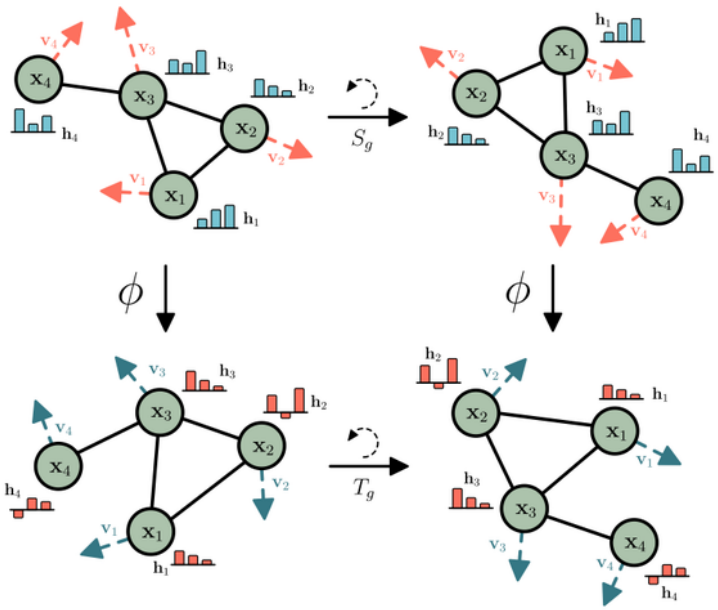


Figure 1: Equivariance depicted by *E(n) Equivariant Graph Neural Networks*

## Conclusion

Mathematical concepts provide powerful tools for modern machine learning applications. From protein structure prediction to graph neural networks and image processing, the principles of equivariance enable to building models that respect the underlying symmetries of real world data. These mathematical foundations continue to drive innovations in the field, demonstrating the enduring value of abstract mathematical concepts in solving real-world problems.

## References


- Emiel Hoogeboom, Victor Garcia Satorras, Max Welling (2021). E(n) Equivariant Graph Neural Networks. ICML 2021.
- Geiger, M., & Smidt, T. (2022). e3nn: Euclidean Neural Networks. arXiv. `https://arxiv.org/abs/2207.09453`
- ICTP Quantitative Life Sciences. (n.d.). Equivariant neural networks and Euclidean symmetry. YouTube. `https://www.youtube.com/watch?v=Ep68XlvfPTI&t=235s`
- Lim, L.-H., Nelson, B. J. (2022). What is an equivariant neural network?. arXiv. `https://arxiv.org/abs/2205.07362`
- Weiler, M. (2023). Equivariant Networks in Deep Learning. `https://maurice-weiler.gitlab.io/blog_post/cnn-book_1_equivariant_networks/`