Symmetry Group Equivariant Neural Networks

Raghav Kansal[1]

[1] *Department of Physics, University of California at San Diego, La Jolla, CA 92093*

I review artificial neural network architectures designed to be equivariant to certain symmetry group transformations. I'll discuss two different but complementary approaches: 'group equivariant convolutional neural networks' and Fourier space decomposition, and focus on $E(2)$, $E(3)$, and $SO(1,3)$ equivariance for their applications in physics.

I. INTRODUCTION

Artificial neural networks have risen to prominence over the last decade during the so-called 'deep learning (DL) revolution'. They have been applied successfully on a variety of computational tasks in fields such as computer vision, natural language processing and even the physical sciences. Particularly in the latter case many datasets, such as molecules and high energy collisions, have intrinsic symmetries like $E(3)$ or $SO(1,3)$, for which it is desirable to develop neural network (NN) architectures which themselves are intrinsically equivariant to the associated transformations. These can be more data efficient, more easily interpretable, and perhaps ultimately, being more naturally suited to the dataset, more successful [1]. In this paper I will, after a brief introduction to the broader field of deep learning, review some proposed NN architectures which are equivariant to various symmetry groups.

II. DEEP LEARNING AND ARTIFICIAL NEURAL NETWORKS

The fundamental building block of all DL models are artificial neural networks (Fig. 1). Modelled vaguely after animal brains, these networks 'learn' synaptic connections, or weights, with which they transform input data into a desired output. Typically in fact the input data is transformed multiple times to intermediate layers, which are ideally learning useful features of the inputs. Networks with large numbers of intermediate layers and hence learnable parameters ('deep' networks) have shown incredible performance in recent years on a vast range of classification, regression and generation tasks.

Some common DL models actually have built-in equivariance to certain transformations. Convolutional neural networks (CNNs) (Fig. 2), which convolve a set of learned filters locally across an input image, and are the industry standard in computer vision, are naturally translation equivariant - translating an input image will commensurately translate the learned feature vectors. Another class of networks, graph neural networks (GNNs) (Fig. 3), act on graphical data and therefore are permutation invariant - permuting input data will not affect the features or output. These are both examples of networks which are consciously designed to respect the symmetries of their respective data and consequently have been extremely successful (especially compared to basic non-equivariant NN architectures). Recently, in the same vein, there has been a large push for NNs equivariant to a broader set of transformations.

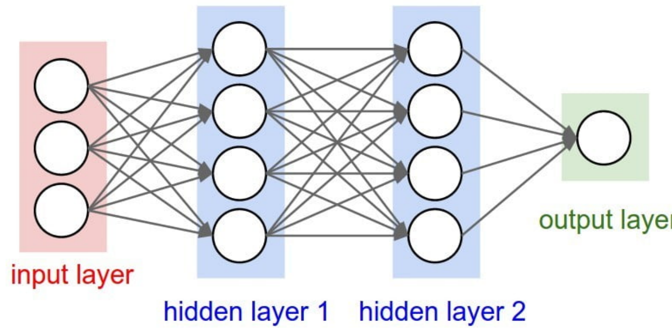

FIG. 1: Artificial Neural Networks

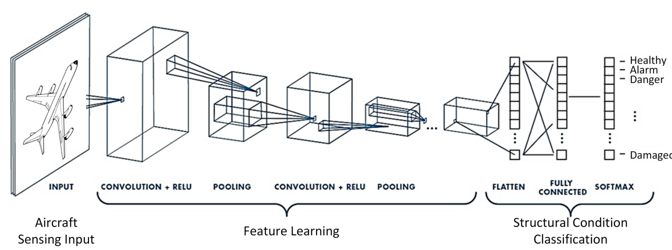

FIG. 2: Convolutional Neural Networks

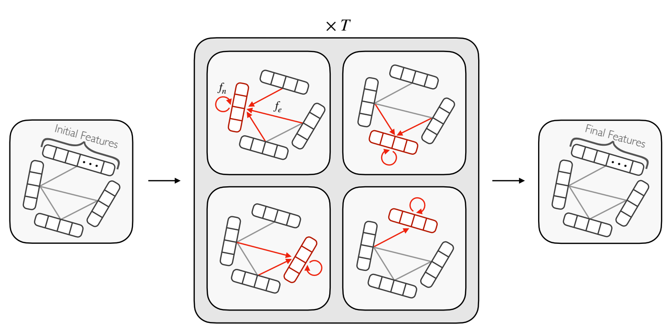

FIG. 3: Graph Neural Networks

III. EQUIVARIANCE

This seems a good point to introduce a very important definition, adapted from Refs. [2–5]: a feature map $f : \mathcal{X} \to \mathcal{Y}$ (e.g. a layer of a network) is considered equivariant to a group of transformations G if $\forall g \in G$ and some representation π there exists a representation π' satisfying

$$\pi'(g)f(x) = f(\pi(g)x) \tag{1}$$

i.e. the group operation commutes with the map f (and f therefore is an intertwiner). Intuitively this means that for each transformation on the input there is a well defined transformation by the same group element in the feature space. Invariance is the particular case where π' is the trivial representation wherein transformations on x do not affect features at all. Refs. [3, 4] argue that equivariance is more desirable in intermediate layers than invariance as it allows learning of useful information about the transformation g itself.

Sec. II mentioned CNNs and GNNs which are equivariant and invariant respectively to $T(N)$ (translations in N dimensions) and S_N (permutations of N objects). Now let us extend these to broader groups.

IV. E(2)

Refs. [2, 3] discuss a general procedure for extending the translational invariance to equivariance to a group $G = T(2) \rtimes H$. where $\rtimes$ is the semi-direct product and H is a subgroup of G, using induced representations. (In this section we will take $G = E(2) \Leftrightarrow H = O(2)$.) The trick is to first find the set of maps $F \ni f$ which satisfy Eq. 1 for an element $h \in H$:

$$\rho_{\text{out}}(h)f = f\rho_{\text{in}}(h) \tag{2}$$

where ρ_{out} and ρ_{in} are reps of H. After this, Eq. 1 can be automatically satisfied using

$$\pi'(g)f = \text{Ind}_H^G(g)f = \rho_{\text{out}}(h)f(\rho_{\text{in}}(h^{-1})(x-t)) \tag{3}$$

where $g = th$ for some $t \in T(2)$.

One simple method for finding F[1] is to recognize that since Eq. 2 is linear in f, all we need is a complete linear basis. An intuitively obvious guess is to have f be a typical CNN layer but with the convolutional filters W_m restricted to circular harmonics [4]:

$$W_m(r, \phi; R, \beta) = R(r)e^{i(m\phi+\beta)} \tag{4}$$

where the radial component R and the filter phase β are learnable parameters. With $m \in \mathbb{Z}$ these filters clearly form a complete basis, and it is a simple matter of a change of integration variables to see that they also satisfy Eq. 2 under convolutions ($*$) with an image $F(r, \phi)$ rotated by θ:

$$W_m * F(r, \phi + \theta) = e^{im\theta}W_m * F(r, \phi) \tag{5}$$

Here ρ_{in} is the fundamental $SO(2)$ rep acting on the image and ρ_{out} is any one of the infinite complex reps. After discretising these filters Ref. [4] demonstrates significant improvement on classification of rotated images compared to state-of-the-art CNNs.

V. E(3)

$E(3)$ equivariance can be achieved in much the same way out of CNNs [6]. Such networks are generally classified as 'group equivariant convolutional neural networks' (G-CNNs) [2]. Let us do something similar to G-CNNs but now also try out 'Fourier decomposition' of the input, feature, and output spaces into irreducible representations (irreps) of the symmetry group.

The most popular such approach for $E(3)$ (Ref. [1]) applies it to datasets of point clouds, which are sets of points in $\mathbb{R}^3$, each with feature vectors in some space $\mathcal{X}$ (so essentially graphs with nodes embedded in $\mathbb{R}^3$). They provide useful representations of physical data such as molecules and crystals, both of which are inherently $E(3)$ invariant.

Each network layer f in such a network must take the set of coordinates $\vec{r}_a$ and features $\vec{x}_a$ and map them to the same set of coordinates with new learnt features $\vec{y}_a$ ($f(\vec{r}_a, \vec{x}_a) = (\vec{r}_a, \vec{y}_a)$, with an equivariant f again having to satisfy our famous Eq. 1.)

Translation equivariance in Ref. [1] is achieved directly by requiring f to only consider distances $\vec{r}_i - \vec{r}_j$ between points i and j (a global translation will not affect these).

For rotation equivariance, first the feature vectors $\vec{x}_a$ are decomposed according to how they transform under irreps of $SO(3)$ - scalars, vectors or higher order tensors (the coordinates are also decomposed the same way but rather obviously they transform under the fundamental rep as vectors):

$$\mathbb{R}^3 \oplus \mathcal{X} = \bigoplus_l R_l^{m_l} \tag{6}$$

where the sum is over irreps R_l (with dimension $2l + 1$) and m_l are the multiplicities. Each point's features and coordinates have the corresponding decomposition:

[1] AKA one offensively hand-wavy method I have improvised in order to reach the same conclusion as of the more rigorous analysis in Refs. [5, 6] without delving into some quite tedious algebra

$$\vec{r}_a \oplus \vec{x}_a = \bigoplus_l \bigoplus_{c=1}^{m_l} V_{ac}^l \tag{7}$$

where the V_{ac}^l are tensors which transform under the l irrep. Each of these tensors are then individually acted upon by generalized convolutional filters with the form $R(r)Y^{l_f}(\hat{r})$, where R is a learnt radial function, Y^l are the spherical harmonic tensors, and the set l_f corresponds to the set of desired irreps in feature space. The spherical harmonics are directly analagous to using circular harmonics for $E(2)$ (except they have dimension $2l+1$) and by the same argument they satisfy Eq. 1. This convolution effectively produces a tensor product representation of $SO(3)$ $R_l \otimes R_{l_f}$, so it is then decomposed via the Clebsch-Gordan (CG) coefficients again into irreps, after which the cycle continues.

A useful pedagogical example is of a network taking as input a collection of pont masses and outputting the moment of inertia tensor. The input features are the masses of each point, which are scalars under $SO(3)$, and the moment of inertia tensor transforms as the $0 \oplus 2$ rep so we define this network to be of the type $0 \to 0 \oplus 2$.

Some more interesting and successful applications include classifying molecules [7], predicting protein complex structures [8], and predicting the phonon density of states (DoS) in crystals [9]. A schematic of the architecture used for the latter is shown in Fig. 4. Different crystals are represented geometrically as point clouds in $\mathbb{R}^3$, with individual atoms labeled via feature vectors $\vec{x}_a$ using mass weighted one-hot encoding. After a series of convolution layers the features are summed over all points to predict 51 scalars comprising the phonon DoS.

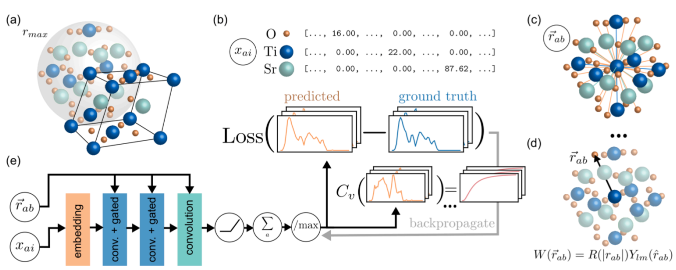

FIG. 4: Schematic of the $E(3)$-equivariant neural network architecture used for predicting phonon DoS.

VI. SO(1, 3)

Recently there has been some success in creating Lorentz group equivariant networks, which are desirable for DL applications to high energy data. There has been no generalization so far of G-CNNs to the Lorentz group[2], but Ref. [10] proposes an alternative, completely Fourier-based, approach, which shares some similarities with our $E(3)$-equivariant network. (Fourier-based here means decomposing into and acting on irreps of a group.)

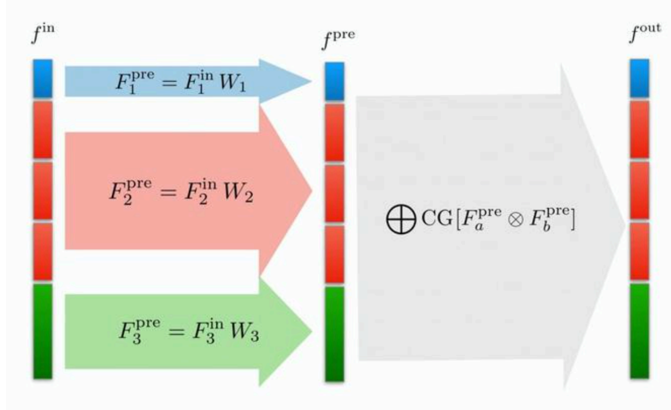

FIG. 5: Schematic of the Lorentz group-invariant network.

The general method is:

1. Decompose the input space into irreps of the group.
2. Apply an equivariant mapping (satisfying Eq. 1) to the feature space.
3. Take tensor products of the irreps and CG decompose them again into irreps.
4. Repeat steps 2-3 until the output layer.

The crucial difference between this and our earlier networks is that the mapping is no longer via convolutional filters; instead, the mapping is chosen to be linear. Recall (Sec. III) that equivariant maps f must be intertwiners between input and output representations, which, according to Schur's Lemma, imposes strong restrictions on both the form of a linear f and its output $f(x)$. Namely: the outputs and inputs must have the same irrep decomposition (although the multiplicities are allowed to vary akin to increasing/decreasing the 'channels' in an image) and f must be a direct sum of learnt matrices acting individually on each irrep. The transformation between f^{in} and f^{pre} in Fig. 5 illustrates such a mapping.

Since this mapping is now linear, we now need to find another way of injecting group equivariant non-linearities into the network[3]. A natural method for doing so is to take tensor products between each pair of irreps after the

[2] As far as I am aware

[3] Non-linearity has been shown to be a necessary ingredient for effective artificial and biological neural networks

mapping and then perform a CG decomposition[4]. Another freedom we have to inject non-linearities is acting with arbitrary learnt functions on any scalar irreps that are produced out of the decomposition since they are, by definition, Lorentz invariants.

One successful application of this network has been to jet tagging, which involves classifying a set of output particles from a particular high energy quark decay (say at the LHC) per the type of quark. Particle features are typically the 4-momenta and possibly scalar features such as particle type. On one such standard dataset, Ref. [10] demonstrates a high (92.9%) accuracy however were unable to match the state-of-the-art using DL (93.8% using a non-Lorentz-equivariant graph CNN [12]).

Finally, note that overall this is in fact a very general approach, applicable to any symmetry group. This includes the aforementioned $E(2)$ and $E(3)$ groups as well as potentially more exotic groups such as E_8 or G_2 which also arise in physics. The only group-dependent operations in such a network are the decompositions into irreps which can readily be calculated for any group (as opposed to G-CNNs where you are required to find group equivariant kernels/convolutional filters).

VII. CONCLUSION

In this paper we reviewed three approaches, involving group-equivariant convolutions and Fourier decompositions, to creating neural networks that are equivariant to certain symmetry groups. Physics datasets often posses intrinsic symmetries so, as demonstrated on some example problems, these networks are promising alternatives/improvements to standard deep learning approaches.

Acknowledgements

Thanks to my classmates Jiashu, Elliot and Ahmed for valuable discussions during the quarter and to Prof. John McGreevy for the amazing course!

[1] N. Thomas et al., "Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds", 2018.

[2] T. S. Cohen and M. Welling, "Group equivariant convolutional networks", 2016.

[3] T. S. Cohen and M. Welling, "Steerable cnns", 2016.

[4] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance", 2017.

[5] M. Weiler and G. Cesa, "General $e(2)$-equivariant steerable cnns", 2019.

[6] M. Weiler et al., "3d steerable cnns: Learning rotationally equivariant features in volumetric data", 2018.

[7] B. K. Miller, M. Geiger, T. E. Smidt, and F. No, "Relevance of rotationally equivariant convolutions for predicting molecular properties", 2020.

[8] S. Eismann et al., "Hierarchical, rotation-equivariant neural networks to predict the structure of protein complexes", *Proteins* (2020).

[9] Z. Chen et al., "Direct prediction of phonon density of states with euclidean neural network", 2020.

[10] A. Bogatskiy et al., "Lorentz group equivariant neural network for particle physics", 2020.

[11] I. Gelfand and R. Minlos, "Representations of Rotation Lorentz Groups and Applications". Elsevier Science & Technology, 1963. ISBN 9780080100692.

[12] H. Qu and L. Gouskos, "Jet tagging via particle clouds", *Phys. Rev. D* **101** (Mar, 2020) 056019, doi:10.1103/PhysRevD.101.056019.

[4] See Ref. [11] for a detailed analysis of CG decomposition for the Lorentz group