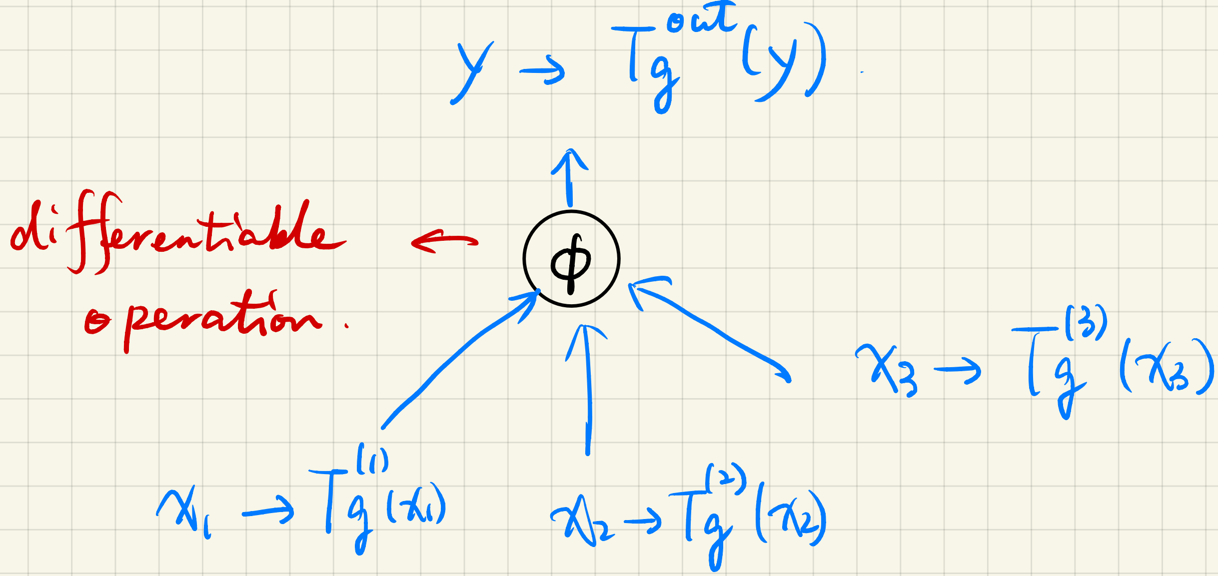
Construction of equivariant neural network

- Incorporating equivariance requires making each "neuron" equivariant to the symmetry group.

- 

$$T_g^{(1)} : \overbrace{x_1 \to x_1' \,,\; T_g^{(2)} : x_2 \to x_2' \,,\; T_g^{(3)}}^{\text{predefined group actions}} : x_3 \to x_3' \cdots$$

A equivariant neuron ϕ has the property that each of its input x_i is transformed based on $T^{(i)}$ of a group G, then its output transforms according to some other action T^{out} of the same group G.

The backpropagation process is automatically equivariant! ③

Now, let's derive the rules for constructing equivariant neurons for any symmetry (as long as the group is compact).

We will use SO(3) group as an example.

"the rotation group in 3D Euclidean space".

Spherical harmonics $\{Y_l^m\}$ are complex valued function defined on the unit sphere.

l: # of angular nodes.

m: orientation of Y_l^m $(-l, \dots, l)$.

Y_l^m appears in Schrödinger Equation's solution for H:

s, p, d ... orbitals $\Leftrightarrow$ $l = 0, 1, 2, \dots$ Y_l^m.

In practice, real-valued spherical harmonics are used in e3nn and similar equivariant neural networks

For our purpose, it's important to know that:

Spherical harmonics are the basis functions for irreducible representations of SO(3) group.

Basic group theory and equivariant network

Group action:

If $x \in \mathbb{R}^3$ and $R \in SO(3)$, then $R \cdot x = Rx$

If $x = (x_1, \dots x_n)$ labels atoms and $\pi \in S_n$, $\pi \cdot x$ reorders the atoms!

So, equivariance is really about group action!

A representation of a group G is a d-dimensional matrix $\rho(g)$ satisfying:

$$\rho(g_2 g_1) = \rho(g_2)\,\rho(g_1) \quad \text{with} \quad g_1, g_2 \in G.$$

A representation is reducible if a basis exists such that:

$$\rho(g) = \begin{pmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{pmatrix} \quad \text{for all } g.$$

If this is impossible, it is an irreducible representation. (irrep)

Peter-Weyl Theorem:

any finite dimensional representation of a compact group G is reducible to a direct sum of irreps of G:

$$\rho(g) = \bigoplus_i \rho_i(g) \text{ , with } \rho_i(g) \text{ as irreps.}$$

$\Rightarrow$ Equivariant operations should be decomposed according to the irreps of the symmetry group!

Now, we can give the neural network operation ("neuron")

$$\phi : (\vec{x}_1, \ldots, \vec{x}_k) \to \vec{y}$$

vectors matching the dimension of $\rho_1, \ldots, \rho_k, \rho_{out}$

a new requirement based on irreps.

The "neuron" is equivariant to G if:

$$\phi\left(\rho_1(g)\vec{x}_1, \rho_2(g)\vec{x}_2, \ldots, \rho_k(g)\vec{x}_k\right) = \rho_{out}(g)\,\phi(\vec{x}_1, \ldots, \vec{x}_k)$$

for all group elements $g \in G$.

Linear equivariant maps

A representation theory called Schur's Lemma:

ρ_1, ρ_2 are two irreps of G, assume that ϕ is a linear function that is equivariant with respect to ρ_1 and ρ_2:

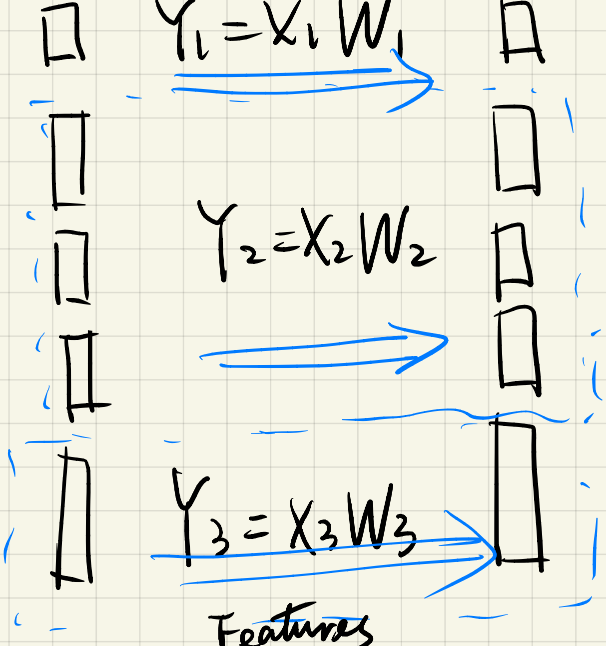
$$\phi(\rho_1(g)\vec{u}) = \rho_2(g)\phi(\vec{u})$$

for all $g \in G$ and all vectors $\vec{u}$, then

If $\rho_1 \neq \rho_2$, then ϕ can only be the zero map $\phi(\vec{u}) = 0$

If $\rho_1 = \rho_2$, then ϕ is a multiple of the identity map.

An equivariant linear neuron cannot mix information from "channels" corresponding to different irreps.



Input and output must be expressed in terms of irreps of the group.

For SO(3) group, the irreps are labelled exactly like angular momentum in QM:

$l = 0, 1, 2, \dots \qquad \dim V_l = 2l+1$

$l=0$: scalar channel.

$l=1$: Vector channel.

$l=2$: quadrupole / rank-2 tensor / ...

Now, we have the linear equivariant layer ready.

Question: how to build nonlinearity?

Regular pointwise nonlinearity breaks equivariance!

We need a new tool for equivariant nonlinearity.

Tensor product

Let's say we have:

vector space

$$\underbrace{\vec{u}}_{\text{vectors}} \in \overbrace{V_{l_1}}^{\text{vector space}}, \qquad \underbrace{\vec{v}}_{\text{vectors}} \in \overbrace{V_{l_2}}^{\text{vector space}}$$

vectors

Tensor product:

$$(\vec{u} \otimes \vec{v})_{ij} = u_i v_j$$ — describe all the pair-wise feature interactions.

Example:

$$\vec{u}, \vec{v} \in \mathbb{R}^3$$

$$\Rightarrow \quad \vec{u} \otimes \vec{v} = \begin{pmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{pmatrix}$$

In reality, it can be decomposed:

$$\vec{u} \otimes \vec{v} = \text{scalar} + \text{antisymmetric} + \text{symmetric}$$

Tensor product of irreps.

If $u \in V_{\ell_1}$, $v \in V_{\ell_2}$, then

Vector space that carries irreps of SO(3).

$$u \otimes \vec{v} \in \underline{V_{\ell_1} \otimes V_{\ell_2}}$$

↳ coupled space.

Decomposition rule:

$$V_{\ell_1} \otimes V_{\ell_2} = \bigoplus_{\ell = |\ell_1 - \ell_2|}^{\ell_1 + \ell_2} V_\ell .$$

decoupled output irrep spaces.

one irrep space

another irrep space

different irrep.

Example: $V_0 \otimes V_1 = V_1$ $\quad$ $V_1 \otimes V_1 = V_0 \oplus V_1 \oplus V_2$

V_0: scalar, V_1: vector, V_1: vector $\quad$ V_1: vector, V_1: vector, V_0: scalar, V_1: vector antisymmetric, V_2: rank-2 symmetric

This is the equivariant way to combine channels!

Dim check: $3 \times 3 = 1 + 3 + 5$.

Clebsch-Gordan coefficients

Let's revisit angular momentum coupling in QM.

$\begin{cases} \text{object 1}: l_1 \\ \text{object 2}: l_2 \end{cases} \Rightarrow l = |l_1 - l_2|, \dots, l_1 + l_2.$

linear combination of uncoupled product states.

$$|lm\rangle = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} |l_1 m_1, l_2 m_2\rangle \underbrace{\langle l_1 m_1, l_2 m_2 | l m\rangle}_{\text{Clebsch-Gordan coefficients (CG)}}$$

coupled state

Clebsch-Gordan coefficients

CG

Now, if we have two equivariant features f^{l_1} and g^{l_2}, the CG decomposition of their tensor product give an output of type L:

CG coefficients

$$h^L_M = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} \langle l_1 m_1, l_2 m_2 | L M\rangle f^{l_1}_{m_1} g^{l_2}_{m_2}$$

valid only for $M = m_1 + m_2$, with $|l_1 - l_2| \le L \le l_1 + l_2$ (CG selection rule)

CG decomposition is general for compact groups.

Different group has different CG coefficients.

Decomposition rule + CG decomposition = equivariant nonlinearity! ⑨

$\underbrace{\text{Decomposition rule}}_{\text{information flow channel}}$ + $\underbrace{\text{CG decomposition}}_{\text{layer output computation}}$

information flow channel   layer output computation.

If f and g each carry multiple ℓ-types, and multiple channels per type (c), the general layer output is:

$$h^{L}_{M,c} = \sum_{\ell_1}\sum_{\ell_2}\sum_{c_1,c_2} W^{L}_{\ell_1\ell_2,c_1c_2,c} \sum_{m_1,m_2} \langle \ell_1 m_1, \ell_2 m_2 | LM\rangle f^{\ell_1,c_1}_{m_1} f^{\ell_2,c_2}_{m_2}$$

ℓ_1, ℓ_2 : input irrep types in f, g.

c_1, c_2 : channel indices within each ℓ type

c : output channel index,

$\langle\ |LM\rangle$: precomputed CG coefficient

$W^{L}_{\ell_1\ell_2,c_1c_2,c}$: learnable weight (depending on $|r|$)

The output of the layer : $h = \bigoplus_{L} h^{L}$

with each h^{L} receives contribution from every valid $(\ell_1, \ell_2) \to L$ path.

Gated nonlinearity : apply regular σ to the scalar channel ($\ell=0$)

Example: $s \in V_0$, $v \in V_1 \Rightarrow V_0 \otimes V_1 = V_1$

"a scalar gate": $v' = \sigma(s)\, v$

- To use scalar feature to gate higher-order features
- Since scalar is invariant, we can use regular σ (sigmoid, ReLU, ...) while keeping equivariance.

Tensor form:

Precompute the CG tensor

$$C^{L}_{l_1 l_2} \in \mathbb{R}^{(2l_1+1)\times(2l_2+1)\times(2L+1)}$$

Then the layer output is:

$$h^{L}_{M,c} = \sum_{l_1 l_2} W^{L}_{l_1,l_2,c} \left(C^{L}_{l_1 l_2}\right)_{m_1 m_2 M} f^{l_1}_{m_1,c_1}\, g^{l_2}_{m_2,c_1}$$