Эквивариантность конволютивных нейросетей относительно групп преобразований входных данных

A.Demichev

March 2021

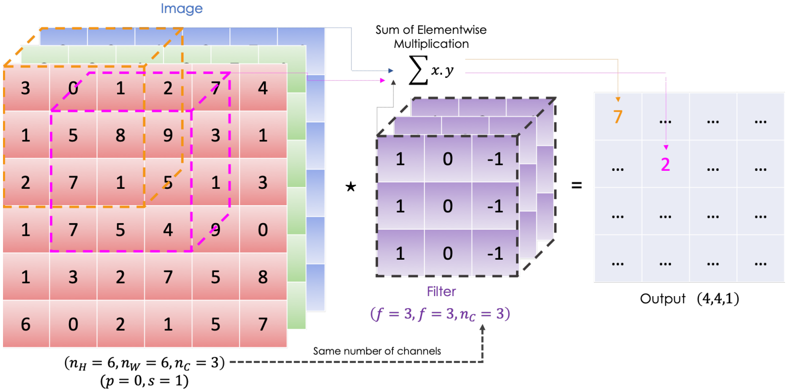
Mainly based on:

- R. Kondor, et al. “On the generalization of equivariance and convolution in neural networks to the action of compact groups.” 2018. ArXiv: 1802.03690
- R. Kondor, et al. “Clebsch–Gordan nets:a fully Fourier space spherical convolutional neural network.” 2018,ArXiv: 1806.09231
  - there are tens of other work on this topic
  - indicated above seems to be most appropriate for the general introduction

Other works:

- T.S. Cohen, M.Geiger, J.Köhler, M.Welling
- S.Ravanbakhsh
- A couple of reviews:
  - C.Esteves "Theoretical aspects of group equivariant neural networks", arXiv:2004.05154
  - L.D.Libera "Deep Learning for 2D and 3D Rotatable Data: An Overview of Methods", arXiv:1910.14594

Classical CNN



- n_H – height; n_W – width; n_C – width
- f – filter size

Classical CNN (2)

- `Input` : $a^{[l-1]}$ with size $(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]})$, $a^{[0]}$ being the image in the input
- `Padding` : $p^{[l]}$, `stride` : $s^{[l]}$
- `Number of filters` : $n_C^{[l]}$ where each $K^{(n)}$ has the dimension: $(f^{[l]}, f^{[l]}, n_C^{[l-1]})$
- `Bias` of the n^{th} convolution: $b_n^{[l]}$
- `Activation function` : $\psi^{[l]}$
- `Output` : $a^{[l]}$ with size $(n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$

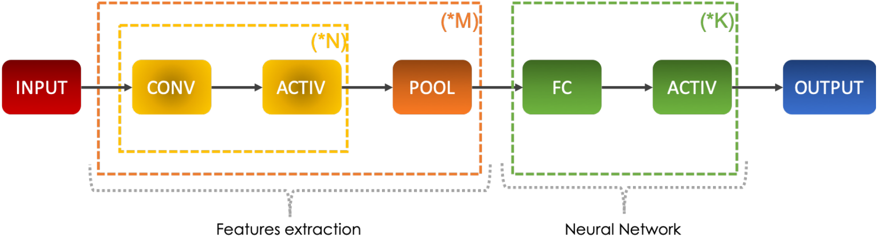
  - мы все эти гиперпараметры рассматривать не будем - только саму конволюцию

$$conv(a^{[l-1]}, K^{(n)})_{x,y} = \psi^{[l]}\left(\sum_{i=1}^{n_H^{[l-1]}} \sum_{j=1}^{n_W^{[l-1]}} \sum_{k=1}^{n_C^{[l-1]}} K_{i,j,k}^{(n)} a_{x+i-1,y+j-1,k}^{[l-1]} + b_n^{[l]}\right)$$

  - но еще (для простоты) опустим смещение
  - я не видел работ, где такие гиперпараметры “восстановлены” для произвольных групп

CNN in Overall

$$a^{[l]} =$$
$$[\psi^{[l]}(conv(a^{[l-1]}, K^{(1)})), \psi^{[l]}(conv(a^{[l-1]}, K^{(2)})), ..., \psi^{[l]}(conv(a^{[l-1]}, K^{(n_C^{[l]})}))]$$
$$dim(a^{[l]}) = (n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$$



Main peculiarities/features of CNNs

- thanks to the convolution they are **equivariant** (*covariant*), including the case of **invariance**;
  - if the input image is translated by any vector (t_1, t_2) (i.e., $f^{0\prime}(x_1, x_2) = f^0(x_1 - t_1, x_2 - t_2)$), then all higher layers will translate in exactly the same way. This property is called **equivariance** (sometimes *covariance*) to translations.
- thanks to the restricted support of the convolution kernel, they are able to generalize details of images
  - the **same filter** is applied to **every part** of the image
    - $\Rightarrow$ if the networks learns to recognize a **certain feature**, e.g., eyes, in one part of the image, then it will be able to do so in **any other part** as well
  - The number of parameters in CNNs is **much smaller** than in fully connected feed-forward networks, since we only have to learn the w^2 numbers defining the χ_ℓ filters rather than $O((m^2)^2)$ weights

Multilayer feed-forward neural network (MFF-NN)

Let $\mathcal{X}_0, \ldots, \mathcal{X}_L$ be a sequence of index sets, $V_0, \ldots, V_L$ vector spaces, $\phi_1, \ldots, \phi_L$ linear maps

$$\phi_\ell \colon L_{V_{\ell-1}}(\mathcal{X}_{\ell-1}) \longrightarrow L_{V_\ell}(\mathcal{X}_\ell),$$

- $L_V(\mathcal{X}) \stackrel{\text{def}}{\equiv}$ the space of functions $\{f \colon \mathcal{X} \to V\}$
- $\sigma_\ell \colon V_\ell \to V_\ell \stackrel{\text{def}}{\equiv}$ appropriate **pointwise nonlinearities**, such as the **ReLU** operator.

The corresponding **multilayer feed-forward NN** is then **a sequence of maps**

$$f_0 \mapsto f_1 \mapsto f_2 \mapsto \ldots \mapsto f_L \ ,$$

where

$$f_\ell(x) = \sigma_\ell(\phi_\ell(f_{\ell-1})(x)). \qquad x \in \mathcal{X}_\ell.$$

- "Flat" neuron indexing is not convenient for consideration of transformations.

Equivariance

Let G be a group and $\mathcal{X}_1, \mathcal{X}_2$ be two sets with corresponding G-actions

$$T_g \colon \mathcal{X}_1 \to \mathcal{X}_1, \qquad T'_g \colon \mathcal{X}_2 \to \mathcal{X}_2.$$

Let V_1 and V_2 be vector spaces, and $\mathbb{T}$ and $\mathbb{T}'$ be the induced actions of G on $L_{V_1}(\mathcal{X}_1)$ and $L_{V_2}(\mathcal{X}_2)$:

$$\mathbb{T}_g \colon f \mapsto f' \qquad f'(x) = f(T_{g^{-1}}(x)).$$

A (linear or non-linear) map $\phi \colon L_{V_1}(\mathcal{X}_1) \to L_{V_2}(\mathcal{X}_2)$ is G-**equivariant** if $\forall\ g \in G$

$$\phi(\mathbb{T}_g(f)) = \mathbb{T}'_g(\phi(f)) \qquad \forall f \in L_{V_1}(\mathcal{X}_1)$$

Equivariance (2)

- Equivariance is represented graphically by a so-called commutative diagram, in this case

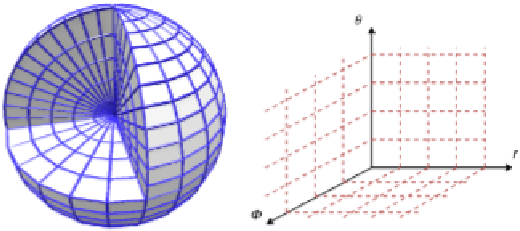
$$\begin{tikzcd} L_{V_1}(\mathcal{X}_1) \arrow[r, "\mathbb{T}_g"] \arrow[d, "\phi"] & L_{V_1}(\mathcal{X}_1) \arrow[d, "\phi"] \\ L_{V_2}(\mathcal{X}_2) \arrow[r, "\mathbb{T}'_g"] & L_{V_2}(\mathcal{X}_2) \end{tikzcd}$$

- Any pointwise functions (non-linearity) are trivially equivariant
  - Movements in $\mathcal{X}$ commute with pointwise transformation of a function

Equivariant feed-forward network

- Let $\mathcal{N}$ be a feed-forward neural network (MFF-NN) and G be a group that acts on each index space $\mathcal{X}_0, \ldots, \mathcal{X}_L$.
- Let $\mathbb{T}^0, \mathbb{T}^1, \ldots, \mathbb{T}^L$ be the corresponding actions on $L_{V_0}(\mathcal{X}_0), \ldots, L_{V_L}(\mathcal{X}_L)$.
- We say that $\mathcal{N}$ is a G–**equivariant feed-forward network** if,
  - when the inputs are transformed $f_0 \mapsto \mathbb{T}^0_g(f_0)$ (for any $g \in G$),
  - the activations of the other layers correspondingly transform as $f_\ell \mapsto \mathbb{T}^\ell_g(f_\ell)$.
- we have not said whether G and $\mathcal{X}_0, \ldots, \mathcal{X}_L$ are discrete or continuous.
  - in certain cases, – when $\mathcal{X}_0 \sim$ sphere or other manifolds which does not have a discretization that fully takes into account its symmetries, it is **easier** to describe the situation in terms of abstract "continuous" neural networks than seemingly simpler discrete
  - in any actual implementation of a neural network, the index sets would of course be finite.
- Note also that invariance is a special case of equivariance, where $T_g = \mathrm{id}$ for all g.

Discretization of a sphere vs. flat spaces



- + **Very** limited number of discrete subgroups of $SO(3)$

Convolution on groups and quotient spaces

- **convolution** of two functions $f, g \colon \mathbb{R} \to \mathbb{R}$

$$(f * g)(x) = \int f(x-y)\, g(y)\, dy. \qquad (1)$$

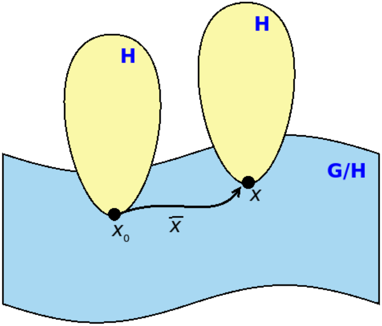
- convolution when f and g are **functions on a compact group** G

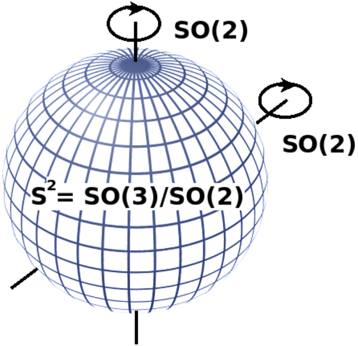
$$(f * g)(u) = \int_G f(uv^{-1})\, g(v)\, d\mu(v). \qquad u, v \in G. \qquad (2)$$

- For discrete groups the integrals are substituted by sums.

Cosets and quotient spaces

- The set of $g \in G$ that map $x_0 \mapsto x$ is a so-called **left coset** $gH := \{gh \mid h \in H\}$.
- The set of all such cosets forms the **(left) quotient space** G/H.
  - $\Rightarrow \mathcal{X}$ can be identified with G/H.
- $\forall$ gH coset we may pick a **coset representative** $g' \in gH$, and let $\overline{x}$ denote the representative of the coset of group elements that map x_0 to x.





Convolution on quotient spaces

- The major complication in neural networks is that $\mathcal{X}_0, \ldots, \mathcal{X}_L$ (spaces that the $f_0, \ldots, f_L$ activations are defined on) are **homogeneous spaces** of G, rather than being G itself.
- Let G be a finite or countable group, $\mathcal{X}$ and $\mathcal{Y}$ be (left or right) quotient spaces of G, $f\colon \mathcal{X} \to \mathbb{C}$, and $g\colon \mathcal{Y} \to \mathbb{C}$.
- We then define the **convolution** of f with g as

$$(f * g)(u) = \sum_{v \in G} f{\uparrow^G}(uv^{-1})\, g{\uparrow^G}(v), \qquad u, v \in G, . \tag{3}$$

- given $f\colon \mathcal{X} \to \mathbb{C}$, we define the **lifting** $f{\uparrow^G}(g) = f(g(x_0)), \quad x = g(x_0)$ for some "origin" x_0.
  - roughly: $f{\uparrow^G}(g)$ is *const* on gH
- Thus in this case: $f * g\colon G \to \mathbb{C}$. In general, this is not what we are looking for.

Convolution on quotient spaces (2)

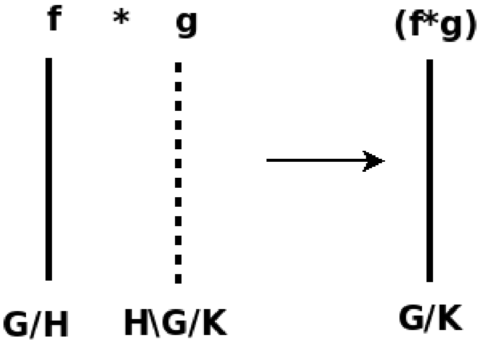
- We need that the convolution on quotient space maps functions on one homogeneous (transitive, quotient) space G/H to another also some homogeneous space G/K, *i.e.* $f \mapsto f * g$ is a map from functions on $\mathcal{X} = G/H$ to functions on $\mathcal{Y} = H/K$. The solution:
- If $f\colon G/H \to \mathbb{C}$, and $g\colon H\backslash G/K \to \mathbb{C}$ then we define the **convolution** of f with g as $f * g\colon G/K \to \mathbb{C}$ with

$$(f * g)(x) = |H| \sum_{y \in H\backslash G} f([\overline{x}\overline{y}^{-1}]_{G/H})\, g([\overline{y}]_{H\backslash G/K}). \qquad (4)$$

  - $x = \overline{x}\, x_0, \qquad y = \overline{y}\, y_0$
  - $[x]_{G/H}$ – projection from G to G/H
- All this looks very complicated. Fortunately, for specific implementations, there are methods that simplify calculations.

Convolution on quotient spaces (3)

Most important:



- Dimensionality: e.g.
  $S^2 = SO(3)/SO(2) \;\rightarrow\; S^2 = SO(3)/SO(2) \Rightarrow$ 1D-filter
  $SO(2)\backslash SO(3)/SO(2)$

Main theorem (Risi Kondor & Shubhendu Trivedi)

- Let G be a compact group and $\mathcal{N}$ be an $L+1$ layer feed-forward neural network
  - in which the ℓ'th index set is of the form $\mathcal{X}_\ell = G/H_\ell$,
  - where H_ℓ is some subgroup of G.
- Then $\mathcal{N}$ is equivariant to the action of G **if and only if** it is a G-CNN.
  - G-CNN: each of the linear maps $\phi_1, \ldots, \phi_L$ in $\mathcal{N}$ is a generalized convolution of the form

$$\phi_\ell(f_{\ell-1}) = f_{\ell-1} * \chi_\ell$$

with some filter $\chi_\ell \in L_{V_{\ell-1} \times V_\ell}(H_{\ell-1} \backslash G / H_\ell)$.

Convolution and Fourier analysis

- the **Fourier transform** of a function f on a (countable) group is defined

$$\widehat{f}(\rho_i) = \sum_{u \in G} f(u)\, \rho_i(u), \qquad i = 0, 1, 2, \ldots, \tag{5}$$

- where $\rho_0, \rho_1, \ldots$ are matrix valued functions called **irreducible representations (irreps)** of G.
- As expected, the generalization of this to the case when f is a function on G/H, $H\backslash G$ or $H\backslash G/K$ is

$$\widehat{f}(\rho_i) = \sum_{u \in G} \rho_i(u)\, f{\uparrow}^G(u), \qquad i = 1, 2, \ldots .$$

  - For details see, e.g. Н.Я. Виленкин "Специальные функции и теория представлений групп"

Convolution theorem on groups

- Let G be a compact group, H and K subgroups of G, and f, g be complex valued functions on G, G/H, $H\backslash G$ or $H\backslash G/K$.
- In any combination of these cases,

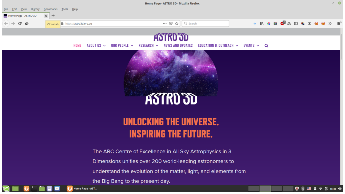
$$\widehat{f * g}(\rho_i) = \widehat{f}(\rho_i)\,\widehat{g}(\rho_i) \tag{6}$$

for any given system of irreps $\mathcal{R}_G = \{\rho_0, \rho_1, \ldots\}$.

- Thus for the Fourier transform the convolution becomes the usual (matrix) multiplication.
- Now we can guess how to **implement** G-CNN if one starts from continuous groups: just cut off the Fourier transform series at L-th term.

Fourier transformed G-CNN on the example of S^2

- The simplest example of possible practical applications – images from cameras on quadcopters
- but also it is possible that it can be applied in astrophysics (?)



- Based on the cited paper by Kondor et al.
  - differs from the pioneering papers by Cohen et al. in a number peculiarities, the main being non-linearities right in Fourier transformed space
  - Cohen et al. perform point-wise nonlinear mapping in **real space** moving **back and forth** between real space and the Fourier domain that comes at a **significant cost** and leads to a range of complications including numerical errors.

Convolutions on the sphere (1)

- On $f^s\colon \mathbb{Z}^2 \to \mathbb{R}$, (with f^0 being the input image), the neurons compute f^s by taking the cross-correlation of the previous layer's output with a small (learnable) filter h^s,

$$(h^s \star f^{s-1})(x) = \sum_y \; h^s(y-x)\, f^{s-1}(y), \tag{7}$$

and then applying a nonlinearity σ, such as the Re-LU operator:

$$f^s(x) = \sigma((h^s \star f^{s-1})(x)). \tag{8}$$

  - cross-correlation differs from the convolution by the order of arguments and despite their name, that is what CNNs actually compute.

Convolutions on the sphere (2)

- On S^2 cross-correlations $h \star f$ is defined as a function *on the rotation group itself*, i.e.,

$$(h \star f)(R) = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi}^{\pi} \left[h_R(\theta, \phi) \right]^* f(\theta, \phi) \cos\theta \, d\theta \, d\phi \quad R \in \mathrm{SO}(3), \tag{9}$$

where h_R is h rotated by R, expressible as

$$h_R(x) = h(R^{-1}x), \tag{10}$$

with x being the point on the sphere at position (θ, ϕ)

- General result by Kondor-Trivedi: if $h, g : G/H \to \mathbb{R}$, $(h \star f) : G \to \mathbb{R}$

Fourier space filters and cross-correlation

- spherical harmonic expansions

$$f(\theta,\phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \widehat{f}_\ell^m Y_\ell^m(\theta,\phi); \quad h(\theta,\phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \widehat{h}_\ell^m Y_\ell^m(\theta,\phi). \tag{11}$$

- Fourier series on the sphere:

$$\widehat{f}_\ell^m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi}^{\pi} f(\theta,\phi)\, Y_\ell^m(\theta,\phi)\, \cos\theta \, d\theta \, d\phi,$$

  and similarly for h.
- for $g\colon \mathrm{SO}(3) \to \mathbb{C}$ the Fourier transform is the collection of *matrices*

$$G_\ell = \frac{1}{4\pi} \int_{\mathrm{SO}(3)} g(R)\, \rho_\ell(R)\, d\mu(R) \qquad \ell = 0, 1, 2, \ldots, \tag{12}$$

  where ρ_ℓ are fixed matrix valued functions = irreducible representations of $\mathrm{SO}(3)$ (Wigner D-matrices).

Fourier space filters and cross-correlation (2)

- Important: Fourier transform of the convolution can be expressed as the outer product of the corresponding $\widehat{f}_\ell$ and $\widehat{h}_\ell^\dagger$ vectors:

$$\widehat{[h \star f]}_\ell = \widehat{f}_\ell \cdot \widehat{h}_\ell^\dagger \qquad \ell = 0, 1, 2, \ldots, L, \tag{13}$$

- Here we used a convenient notation: h_ℓ as $2\ell + 1$-dimensional vector (similarly for any analogous quantities, e.g., f_ℓ)
- Analogously, for functions on $SO(3)$ the resulting cross-correlation formula is almost exactly the same:

$$\widehat{[h \star f]}_\ell = F_\ell \cdot H_\ell^\dagger \qquad \ell = 0, 1, 2, \ldots, L, \tag{14}$$

apart from the fact that now F_ℓ and H_ℓ are matrices

Generalized spherical CNNs

- The central observation: under rotation of input data for a layer
$$\widehat{f}_\ell \mapsto \rho_\ell(R) \cdot \widehat{f}_\ell. \tag{15}$$
$$[\widehat{h \star f}]_\ell \mapsto \rho_\ell(R) \cdot [\widehat{h \star f}]_\ell. \tag{16}$$
$\rho_\ell(R) = \text{Wigner D-matrix}$
- Similarly, if $f', h' \colon \mathrm{SO}(3) \to \mathbb{C}$, then $\widehat{h' \star f'}$ (as defined in (14)) transforms the same way.
- Let $\mathcal{N}$ be an $S+1$ layer feed-forward neural network whose input is a spherical function $f^0 \colon S^2 \to \mathbb{C}^d$.
- We say that $\mathcal{N}$ is a **generalized** $\mathrm{SO}(3)$**–covariant spherical CNN** if the output of each layer s can be expressed as a collection of vectors

$$\widehat{f^s} = (\underbrace{\widehat{f^s_{0,1}}, \widehat{f^s_{0,2}}, \ldots, \widehat{f^s_{0,\tau^s_0}}}_{\ell=0}, \underbrace{\widehat{f^s_{1,1}}, \widehat{f^s_{1,2}}, \ldots, \widehat{f^s_{1,\tau^s_1}}}_{\ell=1}, \ldots\ldots\ldots, \underbrace{\ldots \widehat{f^s_{L,\tau^s_L}}}_{\ell=L}), \tag{17}$$

Generalized spherical CNNs (2)

- here each $\widehat{f}^s_{\ell,j} \in \mathbb{C}^{2\ell+1}$ is a ρ_ℓ–covariant vector in the sense that if the input image is rotated by some rotation R, then $\widehat{f}^s_{\ell,j}$ transforms as
$$\widehat{f}^s_{\ell,j} \mapsto \rho(R) \cdot \widehat{f}^s_{\ell,j}. \tag{18}$$
- We call the individual $\widehat{f}^s_{\ell,j}$ vectors the irreducible **fragments** of $\widehat{f}^s$, and the integer vector $\tau^s = (\tau^s_0, \tau^s_1, \ldots, \tau^s_L)$ counting the number of fragments for each ℓ the **type** (multiplicity (?))of $\widehat{f}^s$.
  - each individual $\widehat{f}^s_\ell$ fragment is effecively a **separate channel**.
- any $\mathrm{SO}(3)$–covariant spherical CNN is equivariant to rotations
- the terms “equivariant” and “covariant” map to the same concept.
  - In this work the term **“equivariant”** is used when one has the same group acting on two objects in a way that is qualitively similar (functions $\leftrightarrow$ cross-correlation). The term **“covariant”** is used if the actions are qualitively different (functions $\leftrightarrow$ irreducible fragments).

Generalized spherical CNNs (3)

- To fully define our neural network, one need to describe three things:
  1. The form of the **linear** transformations in each layer involving learnable weights,
  2. The form of the **nonlinearity** in each layer,
  3. The way that the final output of the network can be reduced to a vector that is rotation **invariant** $\rightarrow$ ultimate goal.

Covariant **linear** transformations

- Let $\widehat{f}^s$ be an $\mathrm{SO}(3)$–covariant activation function of the form (17), and $\widehat{g}^s = \mathcal{L}(\widehat{f}^s)$ be a **linear** function of $\widehat{f}^s$ written in a similar form.
- Then $\widehat{g}^s$ is $\mathrm{SO}(3)$–covariant ***iff*** each $\widehat{g}^s_{\ell,j}$ fragment is a linear combination of fragments from $\widehat{f}^s$ with **the same ℓ**.
  - In other words, it should not entangle irreps with different ℓ.
- With the account of possible multiplicity:

$$G^s_\ell = F^s_\ell \, W^s_\ell \qquad \ell = 0, 1, 2, \ldots, L \tag{19}$$

- the Fourier space cross-correlation formulae (13) and (14) are special cases of (19) corresponding to taking $W_\ell = \widehat{h}^\dagger_\ell$ or $W_\ell = H^\dagger_\ell$.
- The case of general W_ℓ does not have such an intuitive interpretation in terms of cross-correlation.

Covariant nonlinearities: the Clebsch–Gordan transform

- The non-linearity in a real space does not destruct equivariance because of its (non-linearity) point-wise nature
- If one performs the non-linear (namely, quadratic) transformation in the Fourier space, the covariance is provided by the decomposition of the resulting quantity into irreps:
  - Let $\widehat{f}_{\ell_1}$ and $\widehat{f}_{\ell_2}$ be two ρ_{ℓ_1} resp. ρ_{ℓ_2} covariant vectors, and ℓ be any integer between $|\ell_1 - \ell_2|$ and $\ell_1 + \ell_2$. Then

$$\widehat{g}_\ell = C^\top_{\ell_1,\ell_2,\ell}\big[\widehat{f}_{\ell_1} \otimes \widehat{f}_{\ell_2}\big] \tag{20}$$

    is a ρ_ℓ–covariant vector. Here $C_{\ell_1,\ell_2,\ell}$ are the **Clebsch–Gordan** coefficients
- With the account of possible multiplicities the expression becomes a bit more complicated

$$G^s_\ell = \bigsqcup_{|\ell_1 - \ell_2| \le \ell \le \ell_1 + \ell_2} C^\top_{\ell_1,\ell_2,\ell}\big[F^s_{\ell_1} \otimes F^s_{\ell_2}\big], \tag{21}$$

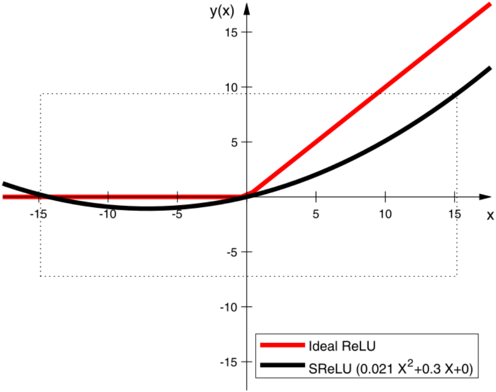
  where $\sqcup$ denotes merging matrices horizontally.
  - This is not important for us for now.

Covariant nonlinearities: justification

- One potential drawback of Cohen's et al. "Spherical CNNs" is that the nonlinear transform in each layer still needs to be computed in "real space".
- $\Rightarrow$ each layer of the network involves a forward and a backward SO(3) Fourier transform,
  - relatively costly
  - is a source of numerical errors, especially since S^2 and $SO(3)$ do not admit any regular discretization
- here **everything** in the Fourier domain
- can it meaningful from the "real space" point of view?

Covariant nonlinearities: justification (2)

- can it meaningful from the "real space" point of view? $\rightarrow$ it seems YES
  - *S.O.Ayat et al. "Spectral-based convolutional neural network without multiple spatial-frequency domain switchings" Neurocomputing 364 (2019) 152–167*
- **main idea:** $c_1(\widehat{f} \star \widehat{f}) + c_2\widehat{f} + c_3 \quad \rightarrow \quad c_1(f \cdot f) + c_2 f + c_3$



Final invariant layer

- After the $S-1$'th layer, the activations of the network will be a series of matrices $F_0^{S-1}, \ldots, F_L^{S-1}$, each transforming under rotations according to $F_\ell^{S-1} \mapsto \rho_\ell(R)\, F_\ell^{S-1}$.
- Ultimately, however, the objective of the network is to output a vector that is *invariant* with respect rotations, i.e., a collection of *scalars*.
  - this simply corresponds to the $\widehat{f}_{0,j}^S$ fragments, since the $\ell = 0$ representation is constant, and therefore the elements of F_0^S are invariant.
- Thus, the **final layer can be similar to the earlier ones, except that it only needs to output this single (single row) matrix**.

Summary of the algorithm

- The **inputs** to the network are n_{in} functions $f_1^0, \ldots, f_{n_{in}}^0 : S^2 \to \mathbb{C}$.
  - E.g., for spherical color images, $f_1^0, f_2^0, f_3^0 =$ red, green and blue
- The activation of layer $s = 0$ is the union of the spherical transforms up to some band limit (resolution) L:

$$[\widehat{f}^0_{\ell,j}]_m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi}^{\pi} f_j^0(\theta,\phi)^* \; Y_\ell^m(\theta,\phi) \cos\theta \, d\theta d\phi. \qquad (22)$$

- For layers $s = 1, 2, \ldots, S-1$, the Fourier space activation $\widehat{f}^s = (F_0^s, F_1^s, \ldots, F_L^s)$

$$G^s_{\ell_1,\ell_2} = F^{s-1}_{\ell_1} \otimes F^{s-1}_{\ell_2} \qquad 0 \leq \ell_1 \leq \ell_2 \leq L. \qquad (23)$$

- decomposing into ρ_ℓ–covariant blocks by

$$[G^s_{\ell_1,\ell_2}]_\ell = C^\dagger_{\ell_1,\ell_2,\ell} \; G^s_{\ell_1,\ell_2}, \qquad (24)$$

Summary of the algorithm (2)

- All $[G^s_{\ell_1,\ell_2}]_\ell$ blocks with the same ℓ are concatenated into a large matrix $H^s_\ell \in \mathbb{C}^{(2\ell+1)\times\overline{\tau^s_\ell}}$, and this is multiplied by the **weight (learnable)** matrix $W^s_\ell \in \mathbb{C}^{\overline{\tau^s_\ell}\times\tau^s_\ell}$ to give

$$F^s_\ell = H^s_\ell \, W^s_\ell \qquad\qquad \ell = 0, 1, \ldots, L. \tag{25}$$

- The operation of the **final** layer S is similar, except that the output type is $\tau^S = (n_{\mathrm{out},0,0,\ldots,0})$, so components with $\ell > 0$ do not need to be computed. By construction, the entries of $F^s_0 \in \mathbb{C}^{1\times n_{\mathrm{out}}}$ are $\mathrm{SO}(3)$–invariant scalars, i.e., they are **invariant** to the simulatenous rotation of the $f^0_1, \ldots, f^0_{n_{in}}$ inputs.
- These scalars may be passed on to a fully connected network or plugged directly into a loss function.

Experiments: Rotated MNIST on the Sphere

- NR/NR = both the training and test sets were not rotated;
- NR/R = the training set was not rotated while the test was randomly rotated;
- R/R = both the training and test sets were rotated

Method	NR/NR	NR/R	R/R
Baseline CNN	97.67	22.18	12
Cohen *et al.*	95.59	94.62	93.4
Kondor *et al.*	96.4	96	96.6

- Other experiments:
  - Atomization Energy Prediction
  - 3D Shape Recognition
- $\rightarrow$ good performance