EECE 517F: Advanced Topics in Deep Learning **Spring 2026**

Lecture 2: Group Convolution

Lecturer: Renjie Liao *Date: Jan 19* *Scribes: Renjie Liao*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*
Disclaimer: *These notes are written with the help from Google Gemini.*

2.1 Group Convolution

Standard Convolutional Neural Networks (CNNs) are designed to be equivariant to translations. Group Convolutional Neural Networks (G-CNNs) generalize this property to larger symmetry groups G, such as rotations and reflections, ensuring that the network's internal representations transform predictably under these actions.

Definition (Discrete Groups) Let G be a group and $f, k : G \to \mathbb{R}$ be functions defined on that group (representing feature maps and kernels, respectively). The Group Convolution is defined as:

$$(f * k)(g) = \sum_{h \in G} f(h) k(g^{-1}h), \tag{2.1}$$

where $g, h \in G$. In the case of discrete groups, the summation is over the group elements; for continuous groups, this becomes an integral over the Haar measure, which will be explained later. Note that $k(g^{-1}h)$ is exactly the left regular representation of the kernel, *i.e.*, $[L_g k](h) = k(g^{-1}h)$ in Eq. (2.1).

2.1.1 Example I: 2D Convolution.

To show how this general definition simplifies to the conventional convolution used in deep learning, we will instantiate the group G as the 2D Translation Group.

The Setup: Translation Group $(\mathbb{Z}^2, +)$: In a standard CNN, we work on a pixel grid. The "symmetries" are just shifts.

- Elements: The elements g and h are 2D integer vectors: $g = (x_g, y_g)$ and $h = (x_h, y_h)$.
- Operation: The group operation is vector addition $(+)$.
- Identity: The zero vector $(0, 0)$.
- Inverse: The inverse of g is $-g = (-x_g, -y_g)$.

The Instantiation: Let's plug these into Eq. (2.1) as follows:

1. Replace g^{-1} with the translation inverse: $-g$.
2. Replace the abstract group product with the translation operation: $+$.
3. The term $g^{-1}h$ becomes: $-g + h$.

Substituting these into the sum:

$$(f * k)(g) = \sum_{h \in \mathbb{Z}^2} f(h) k(h - g)$$

Change of Variables to "standard" Form: In standard image processing, we usually write the kernel relative to a displacement. Let's define a new variable for the "offset" or "distance from the center": $m = h - g$. This implies $h = g + m$. As h sums over the whole grid, m also sums over the whole grid.

$$(f * k)(g) = \sum_{m \in \mathbb{Z}^2} f(g + m)k(m)$$

2.1.2 Equivariance

To prove equivariance of group convolution w.r.t. any group element $u \in G$, we need to show that transforming the input signal f by some group element u results in an identical transformation in the output of the convolution. Mathematically, we use the Left Regular Representation L_u to represent the signals before and after transformations. We want to prove $[L_u(f * k)] = [L_u f] * k$, where we drop the input arguments on both sides for simplicity.

Left-Hand Side (LHS): Transforming the Output. We first compute the convolution and then apply the transformation u to the resulting function.

$$\begin{aligned}
[L_u(f * k)](g) &= (f * k)(u^{-1}g) \\
&= \sum_{h \in G} f(h)k((u^{-1}g)^{-1}h) \\
&= \sum_{h \in G} f(h)k((g^{-1}uh) \qquad (2.2)
\end{aligned}$$

Right-Hand Side (RHS): Transforming the Input. Now, we define a transformed input $f' = [L_u f]$, where $f'(h) = f(u^{-1}h)$, and convolve it with the kernel k.

$$\begin{aligned}
((L_u f) * k)(g) &= ((f' * k)(g) \\
&= \sum_{h \in G} f'(h)k(g^{-1}h) \\
&= \sum_{h \in G} f(u^{-1}h)k(g^{-1}h) \\
&= \sum_{z \in G} f(z)k(g^{-1}uz), \qquad (2.3)
\end{aligned}$$

where we change the variable $z = u^{-1}h$ in the last line. Comparing Eq. (2.2) and Eq. (2.3), we conclude the general equivariance:

$$[L_u(f * k)](g) = [L_u f] * k(g) \qquad (2.4)$$

2.2 Group Convolution in Practice

In practice, group equivariant deep learning models like G-CNNs typcially consists of two operations: the *lifting convolution* and *group-group convolution*, where the former is the first layer and the rest constructs the subsequent layers.

Lifting Convolution ($\mathbb{Z}^2 \to G$): To process a standard image $f : \mathbb{Z}^2 \to \mathbb{R}$, we must first "lift" it into the group domain. For a group G that acts on the plane like $p4$, the lifting convolution is defined as:

$$(f * k)(g) = \sum_{x \in \mathbb{Z}^2} f(x)k(g^{-1}x)$$

where the kernel k is transformed by the group element g. For $p4$, which consists of translations and 90° rotations, the output is a function with 4 orientation channels at every spatial coordinate. In deep learning context, this layer converts a spatial input $[H, W, C_{\text{in}}]$ into a group-based feature map $[H, W, 4, C_{\text{out}}]$. Therefore, we lift the dimension of the representation, that is why this operation is called *lifting convolution*. In group terminologies, the lift convolution moves from the quotient space $\mathbb{Z}^2$ to the full group space C_4 since $\mathbb{Z}^2 \cong p4/C_4$.

Group-Group Convolution ($G \to G$): Once the signal is in the group domain, subsequent layers maintain equivariance by convolving over the entire group structure. For example, in the context of $p4$, we have,

$$(f * k)(g) = \sum_{y \in \mathbb{Z}^2} \sum_{r \in R} f(y, r) k(g^{-1}(y, r))$$

where R is the rotation subgroup. Here, the kernel k is a higher-dimensional object that learns how features at different orientations interact with one another.

Intuition and Practical Considerations:

- Weight Sharing: In a G-CNN, a single learned filter is automatically reused across all group actions (*e.g.*, rotated 4 times in $p4$). This significantly improves parameter efficiency and sample complexity.
- Algebraic vs. Spatial: While the rotational component of $p4$ is algebraically exact on a square grid, the translation component ($\mathbb{Z}^2$) is often an approximation in practice. Because we work with finite $H \times W$ tensors, "exact" equivariance is broken at the boundaries where information "falls off" the grid.
- The Invariant Bottleneck: To produce a final classification that does not depend on the orientation of the input, we typically apply Group Pooling (*e.g.*, $f_{\text{inv}} = \max_{r \in R} f(x, r)$), which collapses the group dimension into a single invariant scalar.

Example $p4$ Group: Let an element $g = (\mathbf{u}, r)$ and $h = (\mathbf{v}, s)$, where $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^2$ are translations and $r, s \in \{0, 1, 2, 3\}$ are rotation indices. Let R_r be the 2×2 rotation matrix corresponding to $r \cdot 90°$. The convolution $(f * k)(g)$ is:

$$(f * k)(\mathbf{u}, r) = \sum_{\mathbf{v} \in \mathbb{Z}^2} \sum_{s=0}^{3} f(\mathbf{v}, s) k\left((\mathbf{u}, r)^{-1}(\mathbf{v}, s)\right)$$

To compute this, we need the $p4$ inverse and group product (the semi-direct product rules). For the ease of notation, let us define $r^{-1} := 4 - r \pmod 4$. We have

1. Inverse: $(\mathbf{u}, r)^{-1} = (-R_{r^{-1}}\mathbf{u}, r^{-1})$.
2. Product: $(\mathbf{a}, r) \cdot (\mathbf{b}, s) = (\mathbf{a} + R_r\mathbf{b}, r + s \pmod 4)$.

Applying these to the kernel index $g^{-1}h$:

$$\begin{aligned}
(\mathbf{u}, r)^{-1}(\mathbf{v}, s) &= (-R_{r^{-1}}\mathbf{u}, r^{-1}) \cdot (\mathbf{v}, s) \\
&= (-R_{r^{-1}}\mathbf{u} + R_{r^{-1}}\mathbf{v}, (4 - r + s) \pmod 4) \\
&= (R_{r^{-1}}(\mathbf{v} - \mathbf{u}), (4 - r + s) \pmod 4) \\
&= (R_r^{-1}(\mathbf{v} - \mathbf{u}), (4 - r + s) \pmod 4),
\end{aligned} \tag{2.5}$$

where we use the fact that $\forall r$, $R_r R_{r^{-1}} = I = R_r R_r^{-1}$ in the last line.

Substituting the result back into the convolution formula, we get the practical implementation used in G-CNNs:

$$(f * k)(\mathbf{u}, r) = \sum_{\mathbf{v} \in \mathbb{Z}^2} \sum_{s=0}^{3} f(\mathbf{v}, s) k\left(R_r^{-1}(\mathbf{v} - \mathbf{u}), (4 + s - r) \pmod 4\right).$$

What is happening here? Spatial Transformation ($R_r^{-1}(\mathbf{v} - \mathbf{u})$): The spatial part of the kernel is rotated by R_r^{-1} and shifted to position $\mathbf{u}$. Channel Transformation ($(4 + s - r) \pmod 4$): This is the "magic" of group convolution. The rotation index r of the output causes a cyclic shift in the rotation channels of the kernel. Example: If the output we are calculating is for $90°$ ($r = 1$), the kernel's $0°$ channel will look at the input's $90°$ channel, the kernel's $90°$ channel will look at the input's $180°$ channel, and so on.

2.3 Continuous Groups and the Haar Measure*

In discrete group convolutions, we compute features by summing over group elements. When generalizing to continuous groups (like $SE(2)$ or $SO(3)$), this summation is replaced by an integral. However, standard integration (Riemann or Lebesgue) is defined for Euclidean spaces. To integrate over a general group G while preserving the group's symmetry structure, we require a special measure: the **Haar Measure**.

2.3.1 Motivation: Invariance

The core property required for Group Equivariance is that the operation must not depend on the specific "frame of reference" of the group. If we shift the integration domain by a group element g, the "volume" of that domain must remain unchanged. Without this property, a shift in the input signal would result in a scaling or distortion of the output features, breaking equivariance.

2.3.2 Formal Definition

Let G be a locally compact topological group. A **Left Haar Measure** is a countably additive, non-trivial measure μ defined on the *Borel subsets* of G that satisfies *left-translation invariance*.

For any Borel subset $S \subseteq G$ and any group element $g \in G$:

$$\mu(gS) = \mu(S) \tag{2.6}$$

where $gS = \{gs \mid s \in S\}$.

Uniqueness Theorem: The Haar measure is unique up to a strictly positive multiplicative constant. This means if μ and ν are both left Haar measures on G, then $\mu = c \cdot \nu$ for some $c > 0$.

2.3.3 Integration on Groups

The invariance condition for sets implies an equivalent invariance for integrals. For any integrable function $f : G \to \mathbb{R}$ and any $g \in G$:

$$\int_G f(h)\, d\mu(h) = \int_G f(gh)\, d\mu(h) \tag{2.7}$$

This property—effectively a "change of variables" rule with a Jacobian of 1—is crucial for proving equivariance.

2.3.4 Examples of Haar Measures

A. Discrete Groups (*e.g.*, $p4$, $\mathbb{Z}^2$) For a discrete group equipped with the discrete topology, the Haar measure is the **Counting Measure**.

$$\mu(S) = |S| \quad \text{(number of elements in } S\text{)} \tag{2.8}$$

The integral becomes a simple summation:

$$\int_G f(h)\, d\mu(h) \equiv \sum_{h \in G} f(h) \tag{2.9}$$

B. The Circle Group (S^1 or $SO(2)$) For the group of continuous planar rotations angles $\theta \in [0, 2\pi)$, the Haar measure is the standard Lebesgue measure on the interval:

$$d\mu(\theta) = d\theta \tag{2.10}$$

Since rotations are compact, we often normalize this measure so that $\mu(G) = 1$, *i.e.*, $d\mu(\theta) = \frac{1}{2\pi} d\theta$.

C. The Special Euclidean Group ($SE(2)$) $SE(2) = \mathbb{R}^2 \rtimes S^1$ consists of translations $\mathbf{x}$ and rotations θ. Since both the translation group $\mathbb{R}^2$ and the rotation group S^1 are *unimodular*[1], the Haar measure on $SE(2)$ is simply the product of their respective measures:

$$d\mu(g) = d\mathbf{x}\, d\theta = dx\, dy\, d\theta \tag{2.11}$$

This means integrating over $SE(2)$ is equivalent to integrating over spatial position and orientation independently.

2.3.5 Continuous Group Convolution

In the continuous setting, the general group convolution is defined using the Haar measure as below:

$$(f * k)(g) = \int_G f(h)k(g^{-1}h)\, d\mu(h) \tag{2.12}$$

Because $d\mu(h)$ is left-invariant, we can prove equivariance by substituting $h' = u^{-1}h$. Since $d\mu(h') = d\mu(h)$, the integral structure is preserved perfectly under transformation.

2.3.6 Proof of Equivariance (The $G \to G$ Case)

Let $f, k : G \to \mathbb{R}$ be square-integrable functions on a unimodular continuous group G (*e.g.*, $SE(2)$).

Theorem: The continuous group convolution commutes with the left regular representation L_u, *i.e.*,

$$[L_u(f * k)] = (L_u f) * k.$$

Proof: First, evaluate the convolution of the transformed input $L_u f$. By definition, $(L_u f)(h) = f(u^{-1}h)$:

$$[(L_u f) * k](g) = \int_G f(u^{-1}h)k(g^{-1}h)\, d\mu(h) \tag{2.13}$$

We perform a change of variables. Let $z = u^{-1}h$.

- This implies $h = uz$.
- Because μ is a **Left Haar Measure**, it is invariant under left translation by u^{-1}. Therefore, the differential element remains unchanged: $d\mu(h) = d\mu(z)$.

Substituting z into the integral:

$$\begin{aligned}(L_u f) * k](g) &= \int_G f(z)k(g^{-1}(uz))\, d\mu(z) \\ &= \int_G f(z)k((u^{-1}g)^{-1}z)\, d\mu(z)\end{aligned}$$

We recognize this expression as the standard convolution $(f * k)$, but evaluated at the point $u^{-1}g$:

$$\begin{aligned}(L_u f) * k](g) &= (f * k)(u^{-1}g) \\ &= [L_u(f * k)](g)\end{aligned}$$

Thus, the operation is equivariant. ∎

[1] A group is called **Unimodular** if the Left Haar Measure is also a Right Haar Measure ($\mu(Sg) = \mu(S)$). For example, compact groups (like $SO(3)$) are always unimodular. Abelian groups (like translations $\mathbb{R}^n$) are always unimodular. $SE(2)$ is unimodular. For Deep Learning, we almost exclusively work with unimodular groups, which simplifies derivations by avoiding the "modular function" $\Delta(g)$ that would otherwise appear when inverting elements.

2.3.7 The Continuous Lifting Convolution ($\mathbb{R}^2 \to SE(2)$)

We also distinguish between the **Group Convolution** (mapping $G \to G$) and the **Lifting Convolution** (mapping $\mathbb{R}^2 \to G$). Both rely on the Haar measure for their mathematical validity. The lifting layer bridges the gap between the spatial input and the group-theoretic internal representation. This is where the dimensionality increase occurs.

Definition: Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the input image and $\psi : \mathbb{R}^2 \to \mathbb{R}$ be a kernel defined on the plane. The lifting convolution lifts the result to the group $SE(2)$ (positions $\mathbf{x}$ and angles θ):

$$(f * \psi)(g) = \int_{\mathbb{R}^2} f(\mathbf{y})\psi(g^{-1}\mathbf{y})\, d\mathbf{y} \tag{2.14}$$

Note that the integral is over the spatial domain $\mathbb{R}^2$ (Lebesgue measure $d\mathbf{y}$), but the output is indexed by $g \in SE(2)$.

How the "Lift" Occurs: Let $g = (\mathbf{x}, \theta)$. The action of g^{-1} on a spatial point $\mathbf{y}$ is given by $g^{-1}\mathbf{y} = R_{-\theta}(\mathbf{y} - \mathbf{x})$. Substituting this into the definition:

$$(f * \psi)(\mathbf{x}, \theta) = \int_{\mathbb{R}^2} f(\mathbf{y})\psi(R_{-\theta}(\mathbf{y} - \mathbf{x}))\, d\mathbf{y} \tag{2.15}$$

Interpretation of the Dimensional Increase

- **Input Space:** $\mathbb{R}^2$. The function f varies only with spatial position $\mathbf{y}$.
- **Output Space:** $\mathbb{R}^2 \times S^1 \cong SE(2)$. The output function depends on both the position $\mathbf{x}$ and the orientation θ.

By fixing θ to a specific angle, say θ_0, we recover a standard spatial convolution with a *rotated filter* $\psi_{\theta_0} = \psi \circ R_{-\theta_0}$. The lifting layer effectively runs a bank of infinitely many rotated filters simultaneously, stacking their outputs along a new axis θ.

$$\text{Output}(\mathbf{x}) \xrightarrow{\text{Lift}} \text{Output}(\mathbf{x}, \theta) \tag{2.16}$$

This transformation from a spatial map to a group (orientation-score) map allows subsequent layers to perform operations purely on the group structure $SE(2)$.