Exploiting Symmetries in Training Convolutional Neural Networks

by Jorge Daniel Agüero Quinteros
Advised by Dr. Robert Kelvey

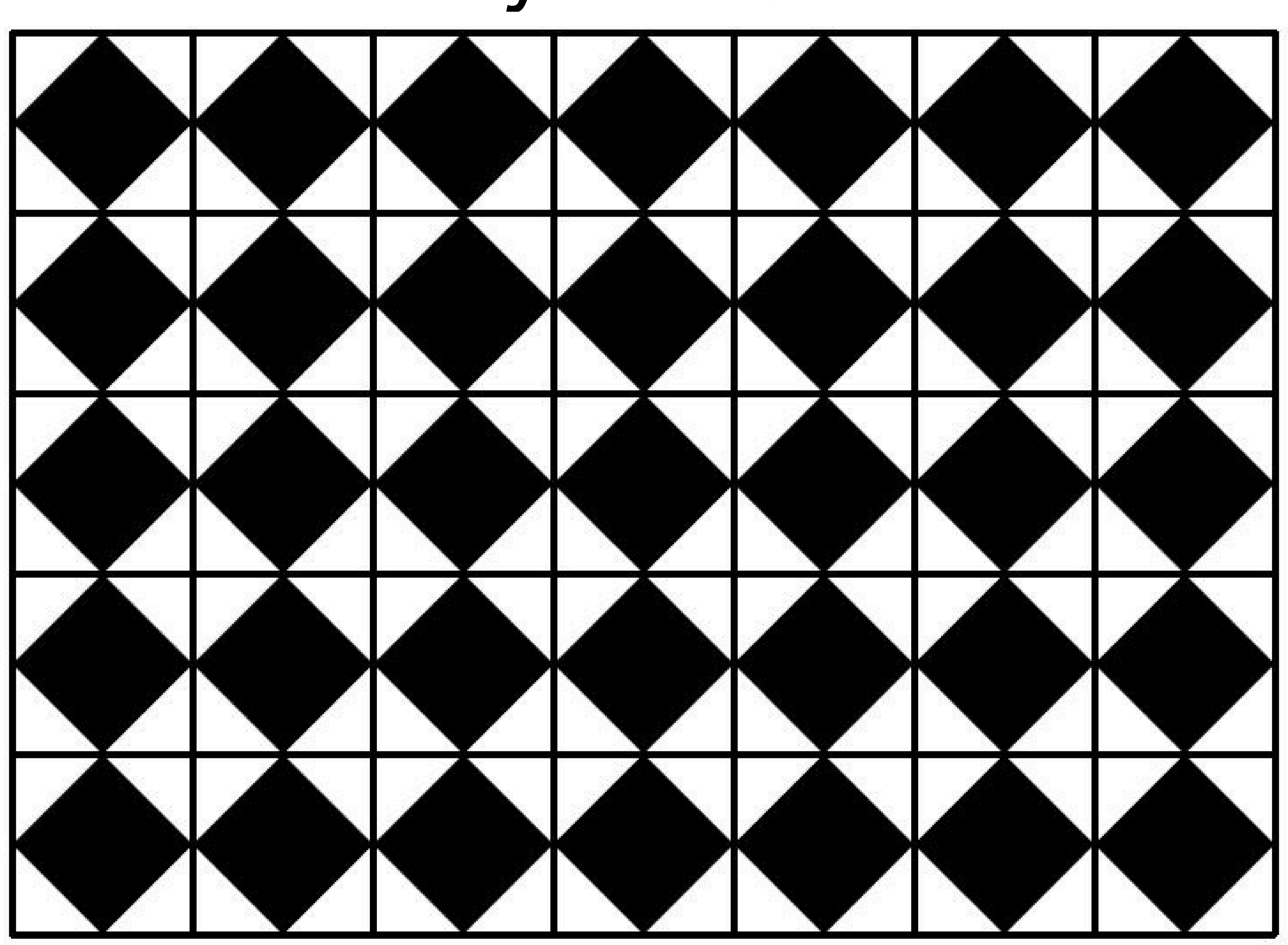
Introduction

Machine Learning models suffer from a few issues:

- Recent models can require millions or billions of parameters, requiring a lot of resources to train.
- The parameters are not human-readable, so we don't know how models work, exactly.
- Models are verified by tinkering and experimentation instead of informed decisions

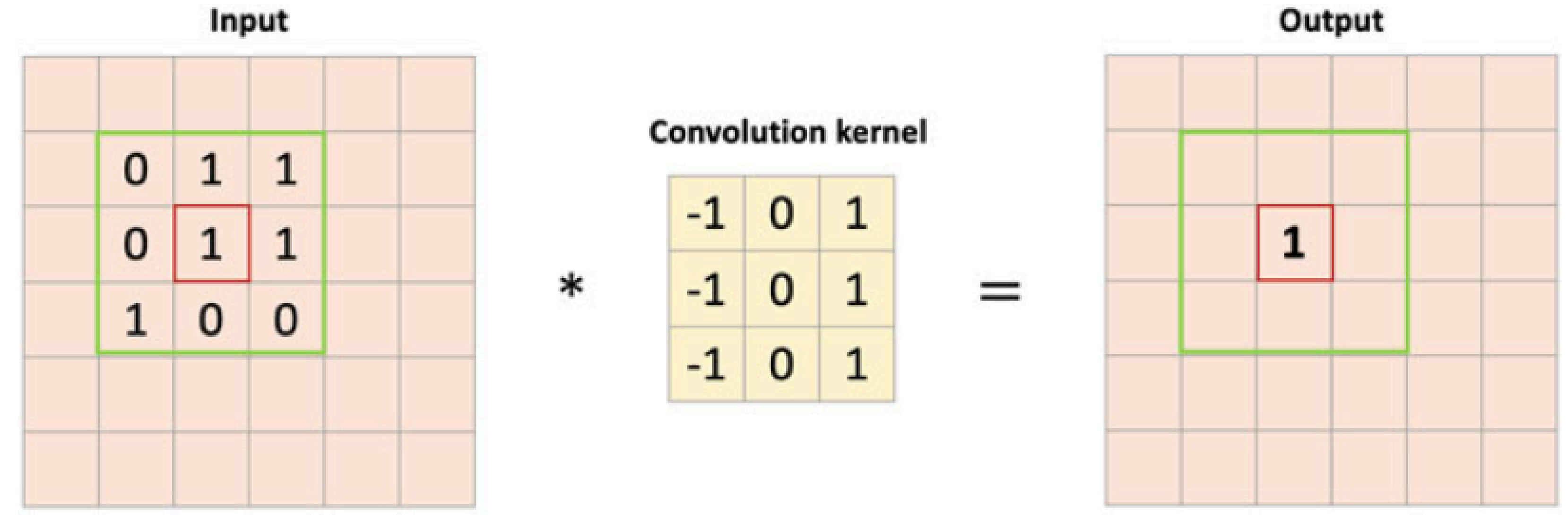
To address these issues, I review and summarize a machine learning model that uses Group Theory to define a model with previous knowledge of symmetries in the dataset

Symmetries



In group theory, we define a **symmetry** as a transformation of a geometric obecjt that leaves it looking the same. For example, the figure above has symmetries of rotations by 90 degrees, reflections , and translations.

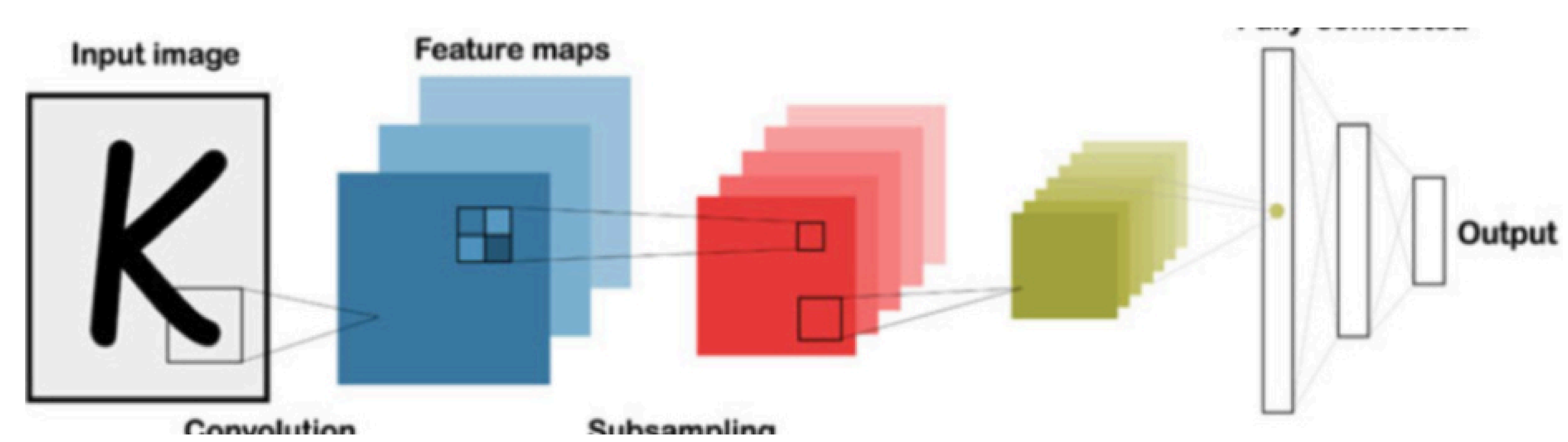
Convolutional Neural Networks



Convolutional Neural Networks (or **CNNs**) produces a **feature map** identifying features in the input:

- CNNs use convolution kernels to compute a single value representing the presence of some feature in a region overlapping with the kernel.
- By translating the kernel across the image and computing values for each region, a convolution produces a feature map.

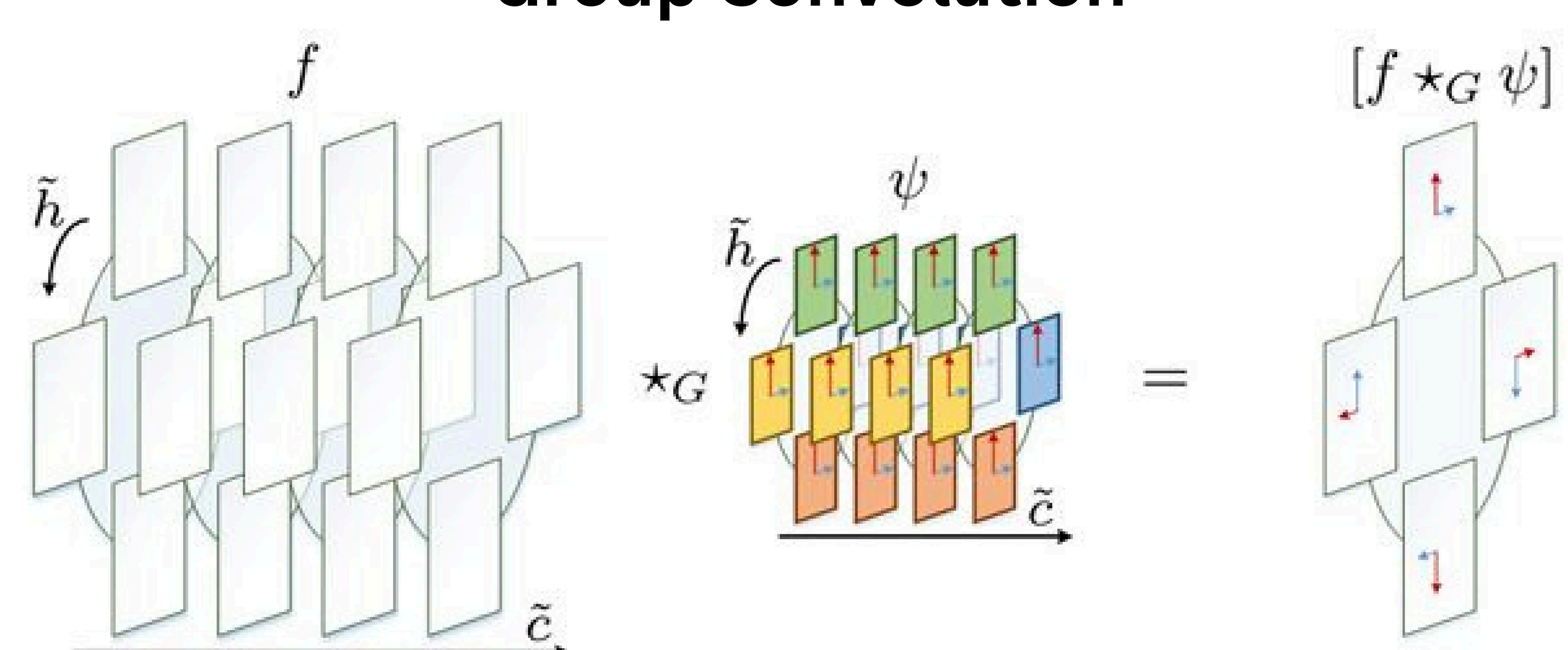
Equivariance of Convolutional Layers



A **Convolutional layer** has many convolution kernels:

- each kernel produces its own feature map
- The whole layer is **equivariant** to translation symmetries: if we translate an input before passing it through the layer, then the output is translated as well.

Group Convolution



A **Group Convolutional Network** (or **GCN**), is an extension of CNNs:

- equivariant to a general class of symmetries called a **group**
- the kernel is applied to regions that are transformed by each symmetry in the specified group. For example, rotating a region and then applying the kernel.

Experimental Results

Model	Error
Z2CNN	5.12%
P4CNNRotationPooling	4.02%
P4CNN	**2.79%**

I reproduced an experiment from 2016 comparing GCNs in classifying rotated hand-written digits:

- Z2CNN uses only translations (like regular CNNs)
- P4CNN use translations and 90-degree rotations
- replaced convolutions in Z2CNN with P4CNN convolutions and kept the same number of parameters
- P4CNN halves the error rate of Z2CNN

Conclusion

GCNs address the previously laid out issues:

- The symmetries act as a multiplier on the number of parameters. That is, less parameters are required for the same performance.
- To use GCNs one must identify symmetries in the dataset, and make the informed decision of which group to use.

Limitations and Future Work

Although GCNs provide an important first step into a generalized equivariant model, it is limited:

- You can only some groups with GCNs
- Parameters are still unreadable
- Experimentation is still needed to verify models

Since its inception, GCNs have been developed in various directions:

- Using infinite and continuous groups
- Convolutions over a sphere instead of a 2D grid
- Other Machine Learning innovations, like transformers or attention.