Group Meeting - April 10, 2020

Paper review & Research progress

Truong Son Hy *

*Department of Computer Science
The University of Chicago

Ryerson Physical Lab

On how to be productive at home (1)

Machiavelli - The Prince

When evening comes, I go back home, and go to my study. On the threshold I **take off my work clothes**, covered in mud and filth, and **put on the clothes an ambassador would wear**. Decently dressed, I enter the ancient courts of rulers who have long since died. There I am warmly welcomed, and I feed on the only food I find nourishing, and was born to savor. I am not ashamed to talk to them, and to ask them to explain their actions. And they, out of kindness, answer me. **Four hours go by without my feeling any anxiety. I forget every worry. I am no longer afraid of poverty, or frightened of death.** I live entirely through them.

On how to be productive at home (2)

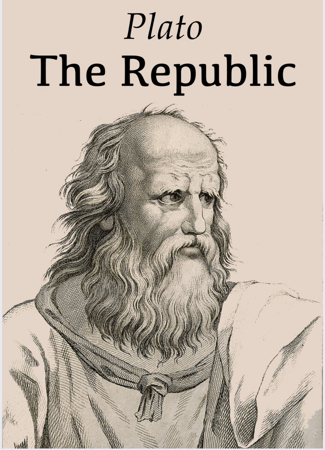
Socrates

No citizen has a right to be an amateur in the matter of **physical training**.

On how to be productive at home (3)

Plato

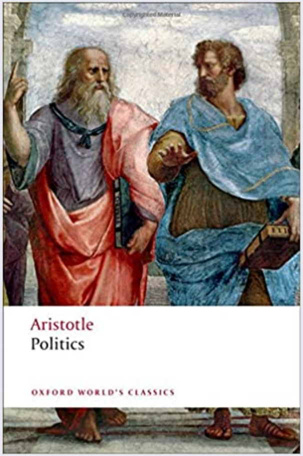
Lack of activitiy destroys the good condition of every human being.



On how to be productive at home (4)

Aristotle

Health is a matter of choice, not a mystery of chance. It is well to be up before daybreak, for such habits contribute to health, wealth and wisdom.



Papers

1. Gauge Equivariant Mesh CNNs Anisotropic convolutions on geometric graphs (preprint) https://arxiv.org/pdf/2003.05425v1.pdf
2. Gauge Equivariant Convolutional Networks and the Icosahedral CNN (ICML 2019) https://arxiv.org/pdf/1902.04615.pdf

Introduction

Convolution on meshes

- A common approach to define convolutions on meshes is to interpret them as a graph and apply graph convolutional networks (GCNs).
- Such GCNs utilize **isotropic** kernels:
    1. Insensitive to the relative orientation of vertices.
    2. Insensitive to the geometry of the mesh as a whole.

Introduction

Convolution on meshes

- A common approach to define convolutions on meshes is to interpret them as a graph and apply graph convolutional networks (GCNs).
- Such GCNs utilize **isotropic** kernels:
  1. Insensitive to the relative orientation of vertices.
  2. Insensitive to the geometry of the mesh as a whole.

Proposal

Gauge Equivariant Mesh CNNs which generalize GCNs to apply **anisotropic** gauge equivariant kernels:

- Resulting features carry orientation information.
- **Geometric message passing**: parallel transporting features over mesh edges.

GCNs approach (1)

A conventional graph convolution between kernel K and signal f, evaluated at a vertex p, can be defined by:

$$(K * f)_p = K_{self} f_p + \sum_{q \in \mathcal{N}_p} K_{neigh} f_q$$

where $\mathcal{N}_p$ is the set of neighbors of p in G, and $K_{self} \in \mathbb{R}^{C_{in} \times C_{out}}$ and $K_{neigh} \in \mathbb{R}^{C_{in} \times C_{out}}$ are the two linear maps which model a self interaction and the neighbor contribution, respectively.

GCNs approach (2)

Problems

- Graph convolution does not distinguish different neighbors, because each feature vector f_q is multiplied by the same matrix K_{neigh} and then summed. For this reason, we say the kernel is **isotropic**.
- A GCNs output at a node p is designed to be independent of relative angles and invariant to any permutation of its neighbors $q_i \in \mathcal{N}(p)$. An isotropic convolution kernel is insensitive to orientation that causes the loss of geometrical information.

Anisotropic kernel (1)

- Direction sensitive (anisotropic) kernels $K(\theta)$ for $\theta \in [0, 2\pi)$ instead of isotropic kernels. Pick an arbitrary reference neighbor q_0^p to determine a reference orientation $\theta_{pq_0^p} = 0$.
- To perform convolution, geometric features at different vertices need to be linearly combined $\rightarrow$ **parallel transport** the feature to the same vertex.
- Apply a matrix $\rho(g_{q \rightarrow p}) \in \mathbb{R}^{C_{out} \times C_{in}}$ to the coefficients of the feature at q to obtain the coefficients of the feature vector transported to p, for the convolution at p.

My thinking

I think this approach is somewhat limited since each mesh face is **planar** (planar rotation group $SO(2)$).

Anisotropic kernel (2)

GEM-CNN convolution:

$$(K * f)_p = K_{self} f_p + \sum_{q \in \mathcal{N}_p} K_{neigh}(\theta_{pq}) \rho(g_{q \to p}) f_q$$

- We require the outcome of the convolution to be equivalent for any choice of reference orientation.
- Thus, we need **anisotropic kernels** which are equivariant under changes of reference orientations (gauge transformations).

Gauge Equivariant Networks (ICML 2019) (1)

Consider signals defined on a manifold M. We define a **gauge** as a position-dependent invertible linear map $w_p : \mathbb{R}^d \to T_pM$, where T_pM is the tangent space of M at p. This determines a frame $w_p(e_1), .., w_p(e_d)$ in $T_p(M)$, where $\{e_i\}$ is the standard frame of $\mathbb{R}^d$.

A **gauge transformation** is a position-dependent change of frame, which can be described by maps $g_p \in GL(d, \mathbb{R})$ (the group of invertiable $d \times d$ matrices). The transformation g_p depends on the position $p \in M$. To change the frame, simply compose w_p with g_p, i.e. $w_p \mapsto w_p g_p$. Component vectors $v \in \mathbb{R}^d$ transform as $v \mapsto g_p^{-1} v$, so that the vector $(w_p g_p)(g_p^{-1} v) = w_p v \in T_pM$ itself invariant.

Gauge Equivariant Networks (ICML 2019) (2)

Exponential map $\exp_p : T_pM \to M$ takes a tangent vector $V \in T_pM$, follows the geodesic (shortest curve) in the direction of V with speed $||V||$ for one unit of time, to arrive at a point $q = \exp_p V$.

For **scalar fields**, we define a filter as a locally supported function $K : \mathbb{R}^d \to \mathbb{R}$. For $q_v = \exp_p w_p(v)$ for $v \in \mathbb{R}^d$, the scalar convolution of K and $f : M \to \mathbb{R}$ at p as follows:

$$(K * f)(p) = \int_{\mathbb{R}^d} K(v) f(q_v) dv$$

Gauge Equivariant Networks (ICML 2019) (3)

For **general fields**, as $f(q_v)$ is transported to p, it undergoes a transformation which will be denoted as $g_{q_v \to p}$. This transformation acts on the feature vector $f(q_v) \in \mathbb{R}^{C_{in}}$ via the representation $\rho_{in}(g_{q_v \to p}) \in \mathbb{R}^{C_{in} \times C_{in}}$. We obtain the generalized convolution:

$$(K * f)(p) = \int_{\mathbb{R}^d} K(v)\rho_{in}(g_{q_v \to p}) f(q_v) dv$$

GEM-CNN convolution:

$$(K * f)_p = K_{self} f_p + \sum_{q \in \mathcal{N}_p} K_{neigh}(\theta_{pq}) \rho(g_{q \to p}) f_q$$

Equivariance imposes a linear constraint on the kernels. We therefore solve for complete sets of **basis-kernels** K^i_{self} and K^i_{neigh} satisfying this constraint and linearly combine them with parameters w^i_{self} and w^i_{neigh} such that:

$$K_{self} = K_i w^i_{self} K^i_{self}$$

$$K_{neigh} = \sum_i w^i_{neigh} K^i_{neigh}$$

The map $\rho : [0, 2\pi) \to \mathbb{R}^{C \times C}$ is called the **type** of the geometric quantity and is known as a group representation of the planar rotation group $SO(2)$. Any feature type can be composed from **irreducible representations** (irreps).

Linear constraint on the kernels (2)

$\rho_{\text{in}} \to \rho_{\text{out}}$	linearly independent solutions for $K_{\text{neigh}}(\theta)$
$\rho_0 \to \rho_0$	$\begin{pmatrix}1\end{pmatrix}$
$\rho_n \to \rho_0$	$\begin{pmatrix}\cos n\theta & \sin n\theta\end{pmatrix}, \begin{pmatrix}\sin n\theta & -\cos n\theta\end{pmatrix}$
$\rho_0 \to \rho_m$	$\begin{pmatrix}\cos m\theta \\ \sin m\theta\end{pmatrix}, \begin{pmatrix}\sin m\theta \\ -\cos m\theta\end{pmatrix}$
$\rho_n \to \rho_m$	$\begin{pmatrix}c_- & -s_- \\ s_- & c_-\end{pmatrix}, \begin{pmatrix}s_- & c_- \\ -c_- & s_-\end{pmatrix}, \begin{pmatrix}c_+ & s_+ \\ s_+ & -c_+\end{pmatrix}, \begin{pmatrix}-s_+ & c_+ \\ c_+ & s_+\end{pmatrix}$
$\rho_{\text{in}} \to \rho_{\text{out}}$	linearly independent solutions for K_{self}
$\rho_0 \to \rho_0$	$\begin{pmatrix}1\end{pmatrix}$
$\rho_n \to \rho_n$	$\begin{pmatrix}1 & 0 \\ 0 & 1\end{pmatrix}, \begin{pmatrix}0 & 1 \\ -1 & 0\end{pmatrix}$

Table 1. Solutions to the angular kernel constraint for kernels that map from ρ_n to ρ_m. We denote $c_\pm = \cos((m \pm n)\theta)$ and $s_\pm = \sin((m \pm n)\theta)$.

Kernel constraint

For any gauge transformation $g \in [0, 2\pi)$ and angle $\theta \in [0, 2\pi)$:

$$K_{neigh}(\theta - g) = \rho_{out}(-g) \cdot K_{neigh}(\theta) \cdot \rho_{in}(g)$$

$$K_{self} = \rho_{out}(-g) \cdot K_{self} \cdot \rho_{in}(g)$$

Non-linearity

Norm non-linearities and gated non-linearities can be used, but generally perform worse in practice compared to point-wise non-linearities.

Proposal: Regular Non-linearity, uses point-wise non-linearities and is approximately gauge equivariant (built on Fourier Transform).

Algorithm 1 Gauge Equivariant Mesh CNN layer

Input: mesh M, input/output feature types $\rho_{\text{in}}, \rho_{\text{out}}$, reference neighbours $(q_0^p \in \mathcal{N}_p)_{p \in M}$.

Compute basis kernels $K_{\text{self}}^i, K_{\text{neigh}}^i(\theta)$ ▷ Sec. 3

Initialise weights w_{self}^i and w_{neigh}^i.

For each neighbour pair, $p \in M, q \in \mathcal{N}_p$: ▷ Sec. 4.

compute neighbor angles θ_{pq} relative to reference neighbor

compute parallel transporters $g_{q \to p}$

Forward (input features $(f_p)_{p \in M}$, weights $w_{\text{self}}^i, w_{\text{neigh}}^i$):

$$f_p' \leftarrow \sum_i w_{\text{self}}^i K_{\text{self}}^i f_p + \sum_{i, q \in \mathcal{N}_p} w_{\text{neigh}}^i K_{\text{neigh}}^i(\theta_{pq}) \rho_{\text{in}}(g_{q \to p}) f_q$$

Experiments

1. Embedded MNIST (rectangle meshes):
    - Isotropic graph CNN: $19.80 \pm 3.43\%$
    - GEM-CNN: $0.60 \pm 0.05\%$
2. Shape correspondence:
    - FAUST dataset: 100 meshes with 80 train and 20 test, human bodies in various positions. 6890 vertices $\rightarrow$ 6890-class segmentation problem.
    - Accuracy: $99.89 \pm 0.02\%$