Invariance of Vector-Cloud Neural Network

Heng Xiao

Virginia Tech

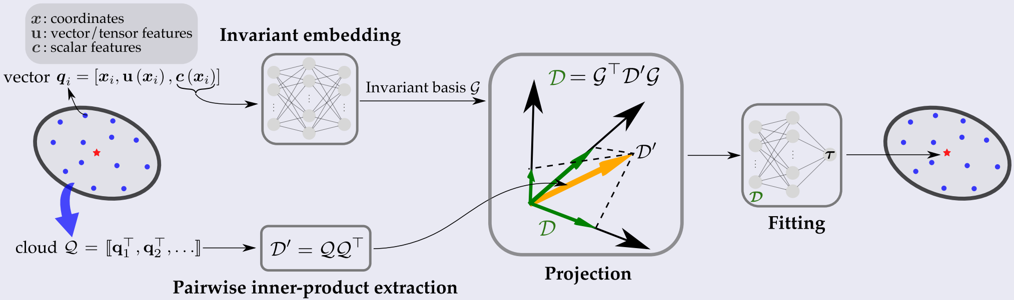
Joint work with:
Xuhui Zhou (Virginia Tech)
Jiequn Han (Princeton)
Ruiying Xu (Delft)

June 1, 2021

Problem Statement

Overview

- We aim to find a mapping from the cloud (⬭) to a cloud center ($\star$).
- The cloud has n points (●), each with a point $\mathbf{q}$ attached to it.
- The cloud is thus $\mathcal{Q} = [\mathbf{q}_1^\top, \mathbf{q}_2^\top, \ldots, \mathbf{q}_n^\top]$.



Nomenclature

Nomenclature (with example values)

- Number of points in cloud: $n = 100$; index i and $i' = 1, \cdots, n$
- Number of features $\ell = 11$; index $j = 1, \cdots, \ell$
- Number of encoding functions $m = 64$, index $k = 1, \cdots, m$

Nomenclature - cont.

Example dimension of variable

- Feature vector for each point: $\mathbf{q} \in \mathbb{R}^{11\times 1}$
- Cloud feature matrix: $\mathcal{Q} \in \mathbb{R}^{100\times 11}$
- Embedding function (64 encoded values for each point) $\mathcal{G} \in \mathbb{R}^{100\times 64}$
- Embedded feature matrix: $\mathcal{G} \in \mathbb{R}^{64\times 64}$
- Reduced embedded matrix $\mathcal{G}^{\star} \in \mathbb{R}^{64\times 4}$ (keeping only the first $m' = 4$ columns for compactness).

Translational and Galilean Invariance

Translational and Galilean invariance is straightforward:

- Use only the *relative* coordinates of the points, not the absolute locations.
- Use only the *relative* velocities (for Galilean invariance).

Rotational Invariance: First Attempt

Pairwise projection among the points can remove dependence on coordinate orientation.

Rotational Invariance with Pairwise Inner Product

- Use only the $\mathbf{q}_i^\top \mathbf{q}_{i'}$, where $i, i' = 1 \dots n$.
- Hence, $\mathcal{D}' = \mathcal{Q}^\top \mathcal{Q}$ is a rotational invariant feature matrix.
- $\mathcal{D}'$ Lacks Permutational Invariance
- Write $\mathcal{D}'_{ii'} = \sum_{j=1}^{\ell} \mathcal{Q}_{ij} \mathcal{Q}_{ji'}$ (i and i' are cloud point indices, and j is feature index).
- $\mathcal{D}'$ would switch rows and columns if the n points are permutated (i.e., the numbering is switched).

How to achieve permutational invariance?

Permutational Invariance - General

Examples

Heuristics: A function where all arguments have the same status is permutational variant, e.g.,

- $f(x_1, x_2, x_3) = \frac{1}{3}(x_1 + x_2 + x_3)$
- $f(x_1, x_2, x_3) = \frac{1}{3}(x_1 + x_2 + x_3) + x_1 x_2 x_3 + x_1 x_2 + x_2 x_3 + x_1 x_3$

Counter-Examples:

These functions are not permutational variant:

- $f(x_1, x_2, x_3) = x_1(x_2 + x_3)$
- $f(x_1, x_2, x_3) = (x_1 + x_2 - x_3)$

The arguments have different status in the function.

Permutational Invariance

Summation over all points can achieve permutational invariance Define a function that depends only on the set of vectors $\{\mathbf{q}_i\}_{i=1}^n$ and .

- Introduce a set of m functions $\{\phi_k(\mathbf{c}_i)\}_{k=1}^m$, where scalars $\mathbf{c}_i$ is part of feature vector $\mathbf{q}_i$, and $k = 1, \ldots, m$ is the embedding function index:

$$\mathcal{L}_{kj} = \frac{1}{n} \sum_{i=1}^{n} \phi_k(\mathbf{c}_i) \mathcal{Q}_{ij}$$

Note

- Inner product (contraction) between ϕ and $\mathcal{Q}$ removed dependence on point ordering (index i, red above).
- This is evident as index i disappeared in the results $\mathcal{L}_{kj}$ after the contraction.

Rotational Invariance: Final Attempt

$\mathcal{L}$ is Permutational Invariant

- Order-removing transformation: $\mathcal{L}_{kj} = \frac{1}{n}\sum_{i=1}^{n} \phi_k(\mathbf{c}_i)\, \mathcal{Q}_{ij}$
- Or in matrix form: $\mathcal{L} = \frac{1}{n} G^\top \mathcal{Q}$ with $\mathcal{G}_{ki} = \phi_k(\mathbf{c}_i)$.

Achieve Rotational Invariance

- $\mathcal{L}$ is obtained by rotating and scaling $\mathcal{Q}$
- $\mathcal{D} = \mathcal{L}\mathcal{L}^\top \equiv \frac{1}{n^2}\mathcal{G}^\top \mathcal{Q}\mathcal{Q}^\top \mathcal{G}$ is rotational invariant
- Note that $\mathcal{D}$ is also translational and permutational invariant because $\mathcal{L}$ is.

Compact Representation

We have transformed cloud feature $\mathcal{Q}$ to invariant $\mathcal{D}$:

$$\mathcal{D} = \mathcal{L}\mathcal{L}^\top \equiv \frac{1}{n^2}\mathcal{G}^\top \mathcal{Q}\mathcal{Q}^\top \mathcal{G}$$

Example Problem Size

- Cloud has $n = 100$ points.
- Each point has a feature vector of length $\ell = 11$.
- We chose $m = 64$ encoding functions, i.e., size of $\phi = \{\phi_j\}_{j=1}^{64}$.
- $\mathcal{Q} \in \mathbb{R}^{100\times 11}$ (1100 raw features)
- $\Longrightarrow \mathcal{D} \in \mathbb{R}^{64\times 64}$ (4096 embedded features)

May not need so many features!

Compact Representation

Feature Compression

- It is desirable to only keep first $m' \ll m$ columns of $\mathcal{D}$ for a more compact encoding.
- To this end, we define:

$$\mathcal{D} = \mathcal{L}\mathcal{L}^{\star\top} \equiv \frac{1}{n^2}\mathcal{G}^\top \mathcal{Q}\mathcal{Q}^\top \mathcal{G}^\star \in \mathbb{R}^{64\times 4}$$

- $\mathcal{G}^\star$ is the first m' columns of $\mathcal{G}$. We used $m' = 4$ in our paper.
- Thus we have $m \times m' = 64 \times 4 = 256$ invariant features.

Overview Again