Equivariant Graph Neural Networks

Kfir Eliyahu Ben Eliav Jonathan Kouchly

December 24, 2024

Outline

1. Motivation
2. Mathematical Background
3. Deep Sets
4. Invariant and Equivariant Graph Networks

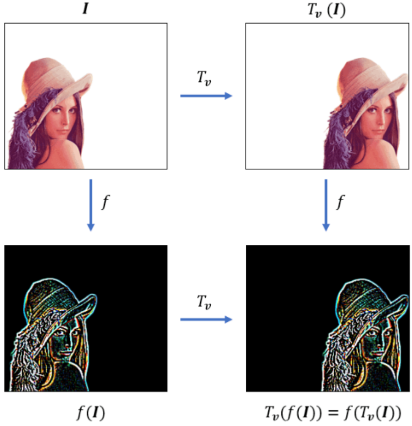
1 Motivation

2 Mathematical Background

3 Deep Sets

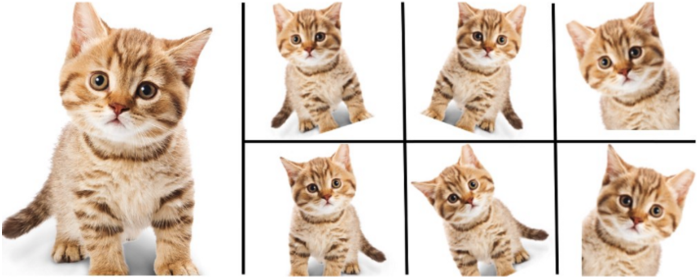
4 Invariant and Equivariant Graph Networks

- Our neural networks can operate on data of many types.
- We often work with images, text, audio, graphs and more.
- These data types have different structures and qualities, and we would like to build architectures that best suit them.



Motivation

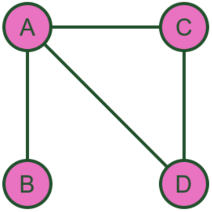
- A cat is a cat no matter how you look at it.



- It is acceptable to assume that being invariant to the rotation of the cat is a good property for a classification network.

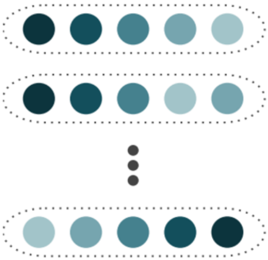
Motivation

- Our focus today is on sets and graph data.



Simple graph

	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	1	0	0	1
D	1	0	1	0



Construction of an Equivariant Neural Network

- When contructing an equivariant neural network, two things should always be considered:
    1. The symmetries of the data:
       What inherent structure should our model be oblivious to?
    2. The space of functions learnable by the network:
       Are we fully utilizing the space of functions that are equivariant

1 Motivation

2 Mathematical Background

3 Deep Sets

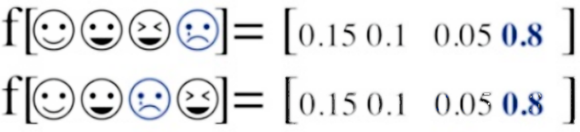
4 Invariant and Equivariant Graph Networks

- The permutation group S_n is the group of all permutations of n elements.
- It has $n!$ elements, representing the $n!$ ways to order n elements.
- Given a set $X = \{x_1, x_2, \ldots, x_n\}$, a permutation $\pi \in S_n$ is a bijection $\pi : X \to X$
- e.g. $x = (x_1, x_2, x_3)$, and $\pi = (1, 2, 3) \in S_3$ is the permutation that maps $1 \to 2$, $2 \to 3$ and $3 \to 1$.
- We denote the **action** of π on x as $\pi x = (x_3, x_1, x_2)$.

Permutation Invariance

- Let $H \leq S_n$ be a subgroup of the symmetric group.

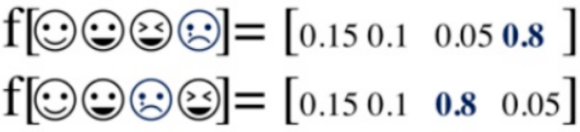
- $f : \mathbb{R}^n \to \mathbb{R}$ is *permutation invariant* if $f(x) = f(\pi x)$ for all $\pi \in H$.



Permutation Equivariance

- Let $H \leq S_n$ be a subgroup of the symmetric group.

- $f : \mathbb{R}^n \to \mathbb{R}^n$ is *permutation equivariant* if $\pi f(x) = f(\pi x)$ for all $\pi \in H$.



Permutation of a Set

- Assume our set is $X = \{x_1, x_2, \ldots, x_n\}$.
- We can represent X as a matrix $X \in \mathbb{R}^{n \times d}$.
- Any permutation $g \in S_n$ can be represented as a permutation matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$,
- The action of g on X is then PX.
- An invariant neural network is a function $f : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{d'}$ such that $f(X) = f(\boldsymbol{P}X)$.
- An equivariant neural network is a function $f : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d'}$ such that $\boldsymbol{P}f(X) = f(\boldsymbol{P}X)$.

Permutation of a Graph

- Our data is now a graph adjacency matric $A \in \mathbb{R}^{n \times n}$.

- A permutation matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ acts on the adjacency matrix A.

- The action of $\boldsymbol{P}$ on A is:

$$\boldsymbol{P}^{\boldsymbol{T}} A \boldsymbol{P}$$

Equivariant Network Construction

Theorem

Let L be a linear equivariant layer, and let f be a neural network constructed be stacking L and non-linearities σ. Then f is permutation equivariant.

Proof.

Let x be a set of n elements, and let $g \in S_n$ be a permutation.

$$f(gx) = L(\sigma(L(\sigma(\ldots L(gx)\ldots)))) = L(\sigma(L(\ldots g\sigma(L(x))\ldots))) = \ldots$$

$$gL(\sigma(L(\sigma(\ldots L(x)\ldots)))) = gf(x)$$

$\square$

Invariant Network Construction

Theorem

Let f be an equvariant neural network and let ϕ be a permutation invariant function. Then $h = \phi(f(x))$ is a permutation invariant neural network.

Proof.

Let x be a set of n elements, and let $g \in S_n$ be a permutation.

$$h(gx) = \phi(f(gx)) = \phi(gf(x)) = \phi(f(x)) = h(x)$$

$\square$

1 Motivation

2 Mathematical Background

3 Deep Sets

4 Invariant and Equivariant Graph Networks

Deep Sets

- A seminal work in the field of equivariant neural networks.
- Recall the two properties we mentioned earlier (symmetries of the data and the space of functions learnable by the network).
- *DeepSets* is an architecture that is equivariant to set permutations and is maximmally expressive in the space of permutation equivariant functions.
- We are going to see the construction and prove it satisfies equivariance and expressiveness.

DeepSets Layer

- We saw a general structure of an invariant and equivariant network.
- To fill in the details, we need to define the equivariant layer L and the invariant function ϕ.

Definition

Consider a set $x = \{x_1, x_2, \dots, x_n\}$, where $x_i \in \mathbb{R}$.
A *DeepSets* layer is defined as

$$L(x) = \lambda \mathrm{I}\mathbf{x} + \mathbf{x1}$$

.

DeepSets Layer

Definition

Consider a set $x = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in \mathbb{R}$.
A *DeepSets* layer is defined as

$$L(x) = \lambda \mathrm{I}\mathbf{x} + \mathbf{x1}$$

.

Theorem

A DeepSets layer is permutation equivariant.

Proof.

Let x be a set of n elements, and let $g \in S_n$ be a permutation.

$$L(gx) = \lambda \mathrm{I}(gx) + (gx)\mathbf{1} = g(\lambda \mathrm{I}x) + x\mathbf{1} = gL(x)$$

$\square$

DeepSets Network

- We have an initial layer L, which we proved is equivariant.
- A deep sets invariant network is now constructed as:

$$f(x) = \phi(L\sigma(L\sigma(\ldots L\sigma(L(x))\ldots))) \quad \text{where} \quad \phi(x) = \sum_{i=1}^{n} x_i$$

- It is easy to see that ϕ is permutation invariant, and thus f is permutation invariant.
- For a classification network, take some classification module ρ (e.g. an MLP), and define the final network as:

$$h(x) = \rho(f(x))$$

DeepSets Network

- Notice that the network is only defined for sets with elements in $\mathbb{R}$.

- We can extend this to a set $X \in \mathbb{R}^{n \times d}$ by defining L as:

$$L(x) = \mathbf{X} W_1 + \mathbf{1}\mathbf{1}^T \mathbf{X} W_2$$

- Ths keeps the general structure of the layer: a linear transformation of the distinct elements of the set summed with the mean of the set.

DeepSets Expressivity

- We have shown that the *DeepSets* network is permutation invariant.

- We now want to show that it is maximally expressive in the space of permutation invariant functions.

Theorem

A DeepSets network is maximally expressive in the space of permutation invariant functions.

Proof.

H.W. (using standard results from approximation theory). □

1 Motivation

2 Mathematical Background

3 Deep Sets

4 Invariant and Equivariant Graph Networks

- We saw a simple, maximally expressive equivariant network for sets.
- We now want to extend this to graphs.

Invariant and Equivariant Graph Networks

- Recall the action of the permutation matrix $\boldsymbol{P}$ on the adjacency matrix A is:

$$\boldsymbol{P}^{\boldsymbol{T}} A \boldsymbol{P}$$

- Lets use this to define a Linear invariant layer $L \in \mathbb{R}^{n \times n}$. We want to satisfy the following:

$$L \boldsymbol{P}^{\boldsymbol{T}} A \boldsymbol{P} = L(A)$$

- But working with bilinear forms is hard. We can define an equivalent condition by column stacking the adjacency matrix to get a vector, and then define a linear layer $L \in \mathbb{R}^{1 \times n^2}$.

$$L\mathrm{vec}(\boldsymbol{P}^{T} A \boldsymbol{P}) = L\mathrm{vec}(A)$$

Invariant and Equivariant Graph Networks

- We now introduce a crucial property of the kronecker product:

$$\text{vec}(XAY) = Y^T \otimes X\text{vec}(A)$$

- Using this, the previous condition can be written as:

$$L(P^T \otimes P^T)\text{vec}(A) = L\text{vec}(A)$$

- For this condition to hold for all A, we need L to satisfy:

$$L(P^T \otimes P^T) = L$$

- transposing the equation, and with severe abuse of notation ($L^T = \text{vec(L)}$), we finally get:

$$(P \otimes P)\text{vec}(L) = \text{vec}(L)$$

Invariant and Equivariant Graph Networks

- After some work and moderate logic jumps, we got a condition for a permutation invariant linear layer L.
- Developing the condition for an equivariant layer is very similar:
-

Conclusion

- end

Thank You!