An Investigation of Lie Groups in Machine Learning

Juncheng Wan, 120033910148, jorgenwan@gamil.com

June 7th, 2021

1 Introduction

As a graduate student of Computer Science and Engineering, I currently focus on machine learning and data mining[1]. In this area, I use machine learning algorithms to approximately solve non-convex problems, such as approximating complex conditional distribution.

One example is **Machine Translation**, which aiming at translating sentence from one language to another language. Specifically, given source sentence $(x_1, \ldots, x_m)$, the machine needs to translate it into the target sentence $(y_1, \ldots, y_n)$, where $x_i, y_j, (1 \leq i \leq m, 1 \leq j \leq n)$ are all random variables. Thus, this problem is equal to approximating the distribution $p(y_1, \ldots, y_n | x_1, \ldots, x_m)$ with parameterized distribution $q_\theta(y_1, \ldots, y_n | x_1, \ldots, x_m)$, where θ indicates the model parameters. However, this problem is difficult for three reasons:

1. The maximal sentence length could be long. For example, in the common translation benchmarks, such as LDC Chinese-English task and WMT14 English-German task, there are sentences with length larger than 512.
2. The vocabulary size could also be large. The vocabulary size of English is from 50,000 to 10,000. Thus, we need to deal with at least 50000^{512} possibilities.
3. The complex phrase composition,grammatical rules, and syntactic structure.

The above problems are not only for machine translation, but also for other natural language processing problems, such as information retrieval, sequence labeling, etc.

From my perspective, utilizing various symmetries of language have two advantages:

1. Reducing the size of the space to be modeled. For example, if there are sentences, the only difference between them is *the place of time adverbials*. Then, I think they are almost equal in meaning, due to the symmetry of the syntactic structure.
2. Providing interpretability of the model. Some machine learning algorithms are considered black-box models and lack interpretability for parameters of the submodule, such as neural networks.

As Lie group is a group of symmetries where the symmetries are continuous, I investigate Lie group in machine learning in this survey out of my interest. I will first give some definitions and concepts of

[1]The website of our lab is `http://apex.sjtu.edu.cn/`.

mathematics nouns and popular machine learning models. Then, I will give some basic results of Lie group in machine learning.

2 Definition

In this section, I clarify basic concepts and definitions.[2]

2.1 Feature Map Vector Space

Definition 1. *Let S be some set, the* ***feature map vector space*** *is defined as:*

$$V \triangleq \{f | f : S \to \mathbb{R}\} \tag{1}$$

First, I want to introduce the concept of feature map. This is a space of scalar-valued functions on the set S, representing the features of each point in S. For example, in computer vision, S could be $\mathbb{R}^2$ which indicating the 2D coordinates and each $f \in V$ could be a gray value function that assigning a gray value for each pixel between $[0, 255]$. In natural language processing, S could be $\{1, 2, \ldots, 512\}$ indicating the discrete positions of words in the sentence and range$(f) \in [50000] \subseteq \mathbb{R}$ indicating the words assignment, where [50000] is the coded vocabulary.

2.2 G-Equivariant/Invariant

Definition 2. *Let G be a group, V_1, V_2 be two feature map vector spaces. A map $\Phi : V_1 \to V_2$ is* ***G-equivariant*** *with respect to actions ρ_1, ρ_2 of G acting on V_1, V_2 respectively if: $\Phi[\rho_1(g)f] = \rho_2(g)\Phi[f]$ for any $g \in G, f \in V_1$.*

Definition 3. *A map $\Phi : V_1 \to V_2$ is* ***G-invariant*** *with respect to actions ρ_1, ρ_2 of G acting on V_1, V_2 respectively if Φ is G-equivariant and ρ_2 is the identity map for any $g \in G$.*

Then, G-equivariant, a concept from representation theory in undergraduate, is necessary. Because in machine learning I am interested in those operation Φ that is equivariant under the group action. For example, there are rotations and transitions in images. I use Φ to map the shallow color features contained in V_1 to useful textural features V_2 and require that they are the same despite the order of rotation or transition of the object.

2.3 Regular Representation

Definition 4. *A* ***regular representation*** *π acting on feature map vector space V is defined as follows:*

$$[\pi(g_\theta)(f)](g_\phi) \triangleq f\left(g_\theta^{-1} g_\phi\right) \tag{2}$$

[2] Though some concepts are fundamental for students of mathematics, they are fresh for me. Thus, I also write them down.

2.4 Lifting

Before introducing lifting, I need to define the homogeneous space:

Definition 5. *Let the group G acts on a set $\mathcal{X}$ via action ρ, if $\forall x, x' \in \mathcal{X}, \exists g \in G : \rho(g)x = x'$, then the action is **transitive** and $\mathcal{X}$ is a **homogeneous space** with respect to G.*

This means that all elements of $\mathcal{X}$ are connected by the action.

Definition 6. *Let $\mathcal{X}$ be a homogeneous space with respect to some group G, $F : \mathcal{X} \to \mathbb{R}$ be a feature map vector space. The **lifting** $\mathcal{L}$ maps $f_{\mathcal{X}} \in F$ (supported on $\bigcup_{i=1}^{n} \{x_i\} \subset \mathcal{X}$) to $\mathcal{L}[f_{\mathcal{X}}]$ (supported on $\bigcup_{i=1}^{n} s(x_i)H \subset G$) such that:*

$$\mathcal{L}\left[f_{\mathcal{X}}\right](g) \triangleq f_{\mathcal{X}}(x_i) \text{ for } g \in s\left(x_i\right) H \tag{3}$$

.

2.5 Lie groups in Machine Learning

I introduce some common Lie groups used in machine learning and their parameterization. Let $\exp : \mathfrak{g} \to G$ and $\log : G \to \mathfrak{g}$ be the Exponential maps and Log maps in the courses. Let the map $\nu : \mathfrak{g} \to \mathbb{R}^d$ represent a function to extracts the free parameters from the Lie algebra element. I give the Lie groups and their corresponding free parameters as follows:

- $G = T(n), t \in \mathbb{R}^n, \nu[\log(t)] = t$.
- $G = SO(2), R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \in \mathbb{R}^{2\times 2}$, $\nu[\log(R)] = \theta = \arctan\left(R_{10}/R_{01}\right)$.
- $G = SE(2), R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \in \mathbb{R}^{2\times 2}, t \in \mathbb{R}^2$, $\nu[\log(tR)] = \begin{bmatrix} t' \\ \theta \end{bmatrix}$, where $t' = V^{-1}t, V = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}, a \triangleq \frac{\sin\theta}{\theta}, b \triangleq \frac{1-\cos\theta}{\theta}$.
- $G = SO(3), R \in \mathbb{R}^{3\times 3}, t \in \mathbb{R}^3$, $\nu[\log(R)] = \nu\left[\frac{\theta}{2\sin\theta}\left(R - R^\top\right)\right] = \frac{\theta}{2\sin\theta}\begin{bmatrix} R_{21} - R_{12} \\ R_{02} - R_{20} \\ R_{10} - R_{01} \end{bmatrix}$, where $\cos\theta = \frac{\mathrm{Tr}(R)-1}{2}$.
- $G = SE(3), R \in \mathbb{R}^{3\times 3}, t \in \mathbb{R}^3$, $\nu[\log(tR)] = \begin{bmatrix} t' \\ r' \end{bmatrix}$, where $t' = V^{-1}t, r' = \nu[\log(R)], V = I + \frac{1-\cos\theta}{\theta^2}\left(R - R^\top\right) + \frac{\theta-\sin\theta}{\theta^3}\left(R - R^\top\right)^2$.

2.6 Lie Convolution Model

First, I define the conventional convolution operation.

Definition 7. *Let $I \in \{f | f : \mathbb{R}^3 \to \mathbb{R}\} \triangleq V_1$ be the feature map of a 2D image with channels (the third dimention), $K \in \{f | f : \mathbb{R}^3 \to \mathbb{R}\} \triangleq V_2$ be a kernel function, n_H, n_W, n_C be the height, width, channels of*

image, feature map vector space $V_3 \triangleq \{f|f:\mathbb{R}^2\to\mathbb{R}\}$. *Then, the* ***convolution*** *operator* $\text{conv}: V_1\times V_2\to V_3$ *is defined as follows:*

$$[\text{conv}(I,K)](x,y)=\sum_{i=1}^{n_H}\sum_{j=1}^{n_W}\sum_{k=1}^{n_C}K(i,j,k)I(x+i-1,y+j-1,k) \tag{4}$$

This operator conv with backpropagation algorithm [5] achieves great improvement in image processing.

As a natural generalization of conventional convolution operation, group equivariant convolution is proposed first in paper [2], which enjoys a substantially higher degree of weight sharing than regular convolution layers.

Definition 8. *The* ***group equivariant convolution*** $\Psi:\mathcal{I}_U\to\mathcal{I}_U$ *is defined as :*

$$[\Psi f](g)\triangleq\int_G\psi\left(g'^{-1}g\right)f\left(g'\right)dg' \tag{5}$$

where $\psi: G\to\mathbb{R}$ is the convolutional filter and the integral is defined withe respect tot the left Haar mearsure of G.

Recently, Lie convolution is proposed, such as papers [3]. The definition is rather similar to the group equivariant convolution. The discretized integral is defined as follows:

$$h\left(u_i\right)=\left(k\hat{*}f\right)\left(u_i\right)=\frac{1}{n_i}\sum_{j\in\text{nbhd}(i)}k\left(v_j^{-1}u_i\right)f\left(v_j\right) \tag{6}$$

where u_i is the group elements, k is the kernel function, $\text{nbhd}(i)=\{v: d(u_i,v)\le r\}$ is the local neighborhood of u_i. The distance $d(u,v)\triangleq\left\|\log\left(u^{-1}v\right)\right\|_F$ is induced by Frobenius norm.

2.7 Lie Self Attention Model

The Self Attention model consists of Scaled Dot-Product Attention and Multi-Head Attention. Assume the input is a sequence of tokens. The Scaled Dot-Product Attention maps linearly those tokens to keys', queries', and values' spaces by matrices Q,K,V. Assume that queries and keys are of dimension d_k, and values of dimension d_v. It does the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and applies a softmax function to obtain the weights on the values. The matrix of Scaled Dot-Product Attention is as:

$$\text{Scaled Dot-Product Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

As it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively, Multi-Head Attention is used. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

$$\begin{aligned}\text{MultiHead}(Q,K,V)=&\ \text{Concat}\ (\ \text{head}_1,\ldots,\ \text{head}_h)\,W^O\\ &\text{where head}_i=\text{Scaled Dot-Product Attention}\left(QW_i^Q,KW_i^K,VW_i^V\right)\end{aligned} \tag{8}$$

where the projections are parameter matrices $W_i^Q\in\mathbb{R}^{d_{\text{model}}\times d_k}, W_i^K\in\mathbb{R}^{d_{\text{model}}\times d_k}, W_i^V\in\mathbb{R}^{d_{\text{model}}\times d_v}$ and $W^O\in\mathbb{R}^{hd_v\times d_{\text{model}}}$.

Lie Self Attention model [6] is using the lifted group elements to substitute the operations between tokens. For example, I can define the followings:

- dot-product: for any two elements $g, g' \in G$ as $\frac{1}{\sqrt{d_v}} \left(W^Q f(g)\right)^\top W^K f\left(g'\right) \in \mathbb{R}$.
- normalization of weights: $\text{softmax}\left(\{\alpha_f\left(g, g'\right)\} g' \in G_f\right)$
- multihead equivariant self-attention:

$$V^m(g) = \int_{G_f} w_f\left(g, g'\right) W^{V,m} f\left(g'\right) dg' \in \mathbb{R}^{d_v/M} \tag{9}$$

From my view, the main difference is that the operation is on group elements, which represent some kind of symmetry.

2.8 Lie Auto Encoder Model

Auto-encoder is a generative neural network model whose encoder compresses the inputs into hidden vectors and the decoder restores the original input. Commonly, the hidden vector is modeled as Gaussian distribution.

Some paper [4] investigate the tangent space of a special Lie group manifold: upper triangular positive definite affine transform matrices (UTDATs). The application is that non-degenerate Gaussian distributions are isomorphic to UTDATs.

As UTDATs form a Lie group, one can work on the tangent spaces (Lie algebras) to make them suitable for machine learning models. Then project back to Lie group by exponential mapping.

3 Main Results

In this section, I collect some results of Lie group in machine learning. My work is more like a research porters.

Theorem 1. *[1] The function composition $f \circ f_K \circ \cdots \circ f_1$ of several equivariant functions $f_k, k \in 1, 2, \ldots, K$ followed by an invariant function f, is an invariant function.*

Proof. Consider group representations $\pi_1, \ldots, \pi_K$ that act on $f_1, \ldots, f_K$ respectively, and representation π_0 that acts on the input space of f_1. If each f_k is equivariant with respect to π_k; π_{k1} such that $f_k \circ \pi_{k1} = \pi_k \circ f_k$, and f is invariant such that $f \circ \pi_k = f$, then we have:

$$\begin{aligned} f \circ f_k \circ \ldots \circ f_1 \circ \pi_0 &= f \circ f_k \circ \ldots \circ \pi_1 \circ f_1 \\ &\vdots \\ &= f \circ \pi_k \circ f_k \circ \ldots \circ f_1 \\ &= f \circ f_k \circ \ldots \circ f_1 \end{aligned} \tag{10}$$

hence $f \circ f_K \circ \cdots \circ f_1$ is invariant. □

Theorem 2. *[6] The group equivariant convolution $\Psi : \mathcal{I}_U \to \mathcal{I}_U$ defined as: $[\Psi f](g) \triangleq \int_G \psi\left(g'^{-1} g\right) f\left(g'\right) dg'$ is equivariant with resepct to the regular representation π of G acting on $\mathcal{I}_U$ as $\left[\pi\left(u\right)\left(f\right)\right]\left(g\right) \triangleq f\left(u^{-1} g\right)$.*

Proof. Use the invariance of the left Haar measure.

$$\begin{aligned}\Psi[\pi(u)f](g) &= \int_G \psi\left(g'^{-1}g\right)[\pi(u)f](g')\,dg' \\ &= \int_{uG} \psi\left(g'^{-1}g\right) f\left(u^{-1}g'\right) dg' \\ &= \int_G \psi\left(g'^{-1}u^{-1}g\right) f(g')\,dg' \\ &= [\Psi f]\left(u^{-1}g\right) \\ &= [\pi(u)[\Psi f]](g)\end{aligned} \tag{11}$$

□

Theorem 3. *The lifting layer $\mathcal{L}$ is equivariant with respect to the representation π.*

Proof. Note $\mathcal{L}[\pi(u)f_{\mathcal{X}}](g) = \mathbf{f}_i$ for $g \in s(ux_i)H$ and $[\pi(u)\mathcal{L}[f_{\mathcal{X}}]](g) = \mathcal{L}[f_{\mathcal{X}}]\left(u^{-1}g\right) = \mathbf{f}_i$ for $g \in us(x_i)H$. Hence $\mathcal{L}[\pi(u)f_{\mathcal{X}}] = \pi(u)\mathcal{L}[f_{\mathcal{X}}]$ because the two cosets are equal: $s(ux_i)H = us(x_i)H, \forall u \in G$. □

In natural language processing, self attention [7] is a powerful model that achieves the best performance in amounts of real-life applications. For more details, please refer to Section 2 or the original paper.

Theorem 4. *[6] LieSelfAttention is equivariant with respect to the regular representation π.*

Proof. Let $\mathcal{I}_U = \mathcal{L}(G, \mathbb{R}^D)$ be the space of unconstrained functions $f : G \to \mathbb{R}^D$. We can define the regular representation π of G acting on $\mathcal{I}_U$ as follows:

$$[\pi(u)f](g) = f(u^{-1}g) \tag{12}$$

f is defined on the set $G_f = \bigcup_{i=1}^n s(x_i)H$ (i.e. union of cosets corresponding to each x_i). Note $G_{\pi(u)f} = uG_f$, and G_f does not depend on the choice of section s.

Note that for all provided choices of k_c and k_l, we have:

$$\begin{aligned}k_c\left([\pi(u)f](g), [\pi(u)f](g')\right) &= k_c\left(f\left(u^{-1}g\right), f\left(u^{-1}g'\right)\right) \\ k_l\left(g^{-1}g'\right) &= k_l\left(\left(u^{-1}g\right)^{-1}\left(u^{-1}g'\right)\right)\end{aligned} \tag{13}$$

Hence for all choices of F, we have that

$$\begin{aligned}\alpha_{\pi(u)f}(g, g') &= F\left(k_c\left([\pi(u)f](g), [\pi(u)f](g')\right), k_l\left(g^{-1}g'\right)\right) \\ &= F\left(k_c\left(f\left(u^{-1}g\right), f\left(u^{-1}g'\right)\right), k_l\left(\left(u^{-1}g\right)^{-1}u^{-1}g'\right)\right) \\ &= \alpha_f\left(u^{-1}g, u^{-1}g'\right)\end{aligned} \tag{14}$$

We thus prove equivariance for the below choice of LieSelfAttention $\Phi : \mathcal{I}_U \to \mathcal{I}_U$ that uses softmax normalisation, but a similar proof holds for constant normalisation. Let $A_f(g, g') \triangleq \exp(\alpha_f(g, g'))$, hence Equation (10) also holds for A_f:

$$\begin{aligned}[\Phi f](g) &= \int_{G_f} w_f(g, g') f(g')\,dg' \\ &= \int_{G_f} \frac{A_f(g, g')}{\int_{G_f} A_f(g, g'')\,dg''} f(g')\,dg'\end{aligned} \tag{15}$$

Hence:

$$\begin{aligned} w_{\pi(u)f}(g,g') &= \frac{A_{\pi(u)f}(g,g')}{\int_{G_{\pi(u)f}} A_{\pi(u)f}(g,g'')\,dg''} \\ &= \frac{A_f\left(u^{-1}g,u^{-1}g'\right)}{\int_{uG_f} A_f\left(u^{-1}g,u^{-1}g''\right)dg''} \\ &= \frac{A_f\left(u^{-1}g,u^{-1}g'\right)}{\int_{G_f} A_f\left(u^{-1}g,g''\right)dg''} \\ &= w_f\left(u^{-1}g,u^{-1}g'\right) \end{aligned} \tag{16}$$

Then we can show that Φ is quivariant with respect to the representation π as follows:

$$\begin{aligned} \Phi[\pi(u)f](g) &= \int_{G_{\pi(u)f}} w_{\pi(u)f}(g,g')\,[\pi(u)f](g')\,dg' \\ &= \int_{uG_f} w_f\left(u^{-1}g,u^{-1}g'\right) f\left(u^{-1}g'\right)dg' \\ &= \int_{G_f} w_f\left(u^{-1}g,g'\right) f(g')\,dg' \\ &= [\Phi f]\left(u^{-1}g\right) \\ &= [\pi(u)[\Phi f]](g) \end{aligned} \tag{17}$$

□

Theorem 5. *[3] Let operator $\mathcal{K}:\mathbb{L}_2(X)\to\mathbb{L}_2(Y)$ be linear and bounded, let X,Y be homogeneous spaces on which Lie group G act transitively, and $d\mu_X$ a Radon measure on X, then:*

1. *K is a kernel operator, i.e., $\exists_{\tilde{k}\in\mathbb{L}_1(Y\times X)}:(\mathcal{K}f)(y)=\int_X \tilde{k}(y,x)f(x)\mathrm{d}\mu_X$*
2. *under the G-equivariance constraint of Eq. (3) the map is defined by a one-argument kernel:*

$$\tilde{k}(y,x)=\frac{\mathrm{d}\mu_X\left(g_y^{-1}\odot x\right)}{\mathrm{d}\mu_X(x)}k\left(g_y^{-1}\odot x\right) \tag{18}$$

   for any $g_y\in G$ such that $y=g_y\odot y_0$ for some fixed origin $y_0\in Y$

3. *if $Y\equiv G/H$ is the quotient of G with $H=\mathrm{Stab}_G(y_0)=\{g\in G\mid g\odot y_0=y_0\}$ then the kernel is constrained via:*

$$\forall_{h\in H},\forall_{x\in X}:\quad k(x)=\frac{\mathrm{d}\mu_X\left(g_y^{-1}\odot x\right)}{\mathrm{d}\mu_X(x)}k\left(h^{-1}\odot x\right) \tag{19}$$

Proof. 1. It follows from Dunford-Pettis Theorem, that if $\mathcal{K}$ is linear and bounded it is an integral operator.

2. The left-equivariance constraint then imposes bi-left-invariance of the kernel $\tilde{k}$ as follows, where $\forall_{g\in G}$ and $\forall_{f\in\mathbb{L}_2(X)}$:

$$\begin{aligned}
&\left(\mathcal{K}\circ\mathcal{L}_g^{G\to\mathbb{L}_2(X)}\right)(f)=\left(\mathcal{L}_g^{G\to\mathbb{L}_2(Y)\circ\mathcal{K}}\right)(f)\quad\Leftrightarrow\\
&\int_X\tilde{k}(y,x)f\left(g^{-1}x\right)\mathrm{d}x=\int_X\tilde{k}\left(g^{-1}y,x\right)f(x)\mathrm{d}x\overset{\text{in r.h.s. integral}}{\Leftrightarrow}x\leftarrow g^{-1}x\\
&\int_X\tilde{k}(y,x)f\left(g^{-1}x\right)\mathrm{d}x=\int_X\tilde{k}\left(g^{-1}y,g^{-1}x\right)f\left(g^{-1}x\right)\mathrm{d}\left(g^{-1}x\right)\Leftrightarrow\\
&\int_X\tilde{k}(y,x)f\left(g^{-1}x\right)\mathrm{d}x=\int_X\tilde{k}\left(g^{-1}y,g^{-1}x\right)f\left(g^{-1}x\right)\frac{1}{|\det g|}\mathrm{d}x
\end{aligned}\tag{20}$$

Since the final equation holds for all $f\in\mathbb{L}_2(X)$ we obtain:

$$\forall_{g\in G}:\quad\tilde{k}(y,x)=\frac{1}{|\det g|}\tilde{k}\left(g^{-1}y,g^{-1}x\right)\tag{21}$$

Furthermore, since G acts transitively on Y we have that $\forall_{y,y_0\in Y}\exists_{g_y\in G}$ such that $y=g_yy_0$ and thus

$$\tilde{k}(y,x)=\tilde{k}\left(g_yy_0,x\right)=\frac{1}{|\det g_y|}\tilde{k}\left(y_0,g_y^{-1}x\right)=:\frac{1}{|\det g_y|}k\left(g_y^{-1}x\right)\tag{22}$$

for every $g_y\in G$ such that $y=g_yy_0$ with arbitrary fixed origin $y_0\in Y$.

3. Every homogeneous space Y of G can be identified with a quotient group $G=H$. Choose an origin $y_0\in Y$ s.t. $\forall_{h\in H}:hy_0=y_0$, i.e., $H=\mathrm{Stab}_G\,y_0$, then

$$\tilde{k}\left(y_0,x\right)=\tilde{k}\left(hy_0,x\right)\Leftrightarrow k(x)=\frac{1}{|\det h|}k\left(h^{-1}x\right)\tag{23}$$

□

Theorem 6. *[4] Let $\boldsymbol{G}_i$ be the UTDAT and $\boldsymbol{g}_i$ be the corresponding vector in its tangent Lie algebra at the standard Gaussian. Then:*

$$\boldsymbol{G}_i=\begin{bmatrix}\sigma_{i1}&&&\mu_{i1}\\&\sigma_{i2}&&\mu_{i2}\\&&\ddots&\vdots\\0&0&\dots&1\end{bmatrix}\tag{24}$$

$$\boldsymbol{g}_i=\begin{bmatrix}\phi_{i1}&&&\theta_{i1}\\&\phi_{i2}&&\theta_{i2}\\&&\ddots&\vdots\\0&0&\dots&1\end{bmatrix},\tag{25}$$

where:

$$\phi_{ik}=\log\left(\sigma_{ik}\right)\tag{26}$$

$$\theta_{ik}=\frac{\mu_{ik}\log\left(\sigma_{ik}\right)}{\sigma_{ik}-1}\tag{27}$$

Proof. By the definition of UTDAT, we can straightforwardly get the first equation.

Let $\boldsymbol{H} = \boldsymbol{G}_i - \boldsymbol{I}$, Using the series form of matrix logarithm, we have:

$$\begin{aligned}\boldsymbol{g}_i &= \log\left(\boldsymbol{G}_i\right) \\ &= \log(\boldsymbol{I} + \boldsymbol{H}) \\ &= \sum_{t=1}^{\infty} (-1)^{t-1} \frac{\boldsymbol{H}^t}{t}.\end{aligned} \tag{28}$$

By substituting $\boldsymbol{H}$ into it, we get the second equation and the following:

$$\begin{aligned}\phi_{ik} &= \sum_{t=1}^{\infty} (-1)^{t-1} \frac{\left(\sigma_{ik} - 1\right)^t}{t} \\ &= \log\left(\sigma_{ik}\right)\end{aligned} \tag{29}$$

and:

$$\begin{aligned}\theta_{ik} &= \sum_{t=1}^{\infty} (-1)^{t-1} \frac{\mu_{ik}\left(\sigma_{ik} - 1\right)^{t-1}}{t} \\ &= \frac{\mu_{ik} \log\left(\sigma_{ik}\right)}{\sigma_{ik} - 1}\end{aligned} \tag{30}$$

Alternatively, after we identify $\boldsymbol{g}_i$ has the form as in the second equation, we can derive the exponential mapping by the definition of matrix exponential:

$$\begin{aligned}\boldsymbol{G}_i = \exp\left(\boldsymbol{g}_i\right) &= \sum_{t=0}^{\infty} \frac{\boldsymbol{g}_i^t}{t!} \\ &= \begin{bmatrix} \sum_{t=0}^{\infty} \frac{\phi_{i1}^t}{t!} & & \theta_{i1} \sum_{t=1}^{\infty} \frac{\phi_{i1}^{t-1}}{t!} \\ & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \\ &= \begin{bmatrix} e^{\phi_{i1}} & & \frac{\theta_{i1}}{\phi_{i1}} \left(\sum_{t=0}^{\infty} \frac{\phi_{i1}^t}{t!} - 1 \right) \\ & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \\ &= \begin{bmatrix} e^{\phi_{i1}} & & \frac{\theta_{ik}\left(e^{\phi_{i1}} - 1\right)}{\phi_{i1}} \\ & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix}.\end{aligned} \tag{31}$$

□

4 Conclusion

To sum up, in this project, I investigate the applications of Lie group in machine learning. The applications are mainly in the improvement of convolution operator in computer vision, self attention operator in natural language processing, and auto-encoder. Most of the time, I am like a Theorem porter. In my eyes, I need to see the code for better understanding the detailed implementation.

References

[1] Benjamin Bloem-Reddy and Yee Whye Teh. "Probabilistic Symmetries and Invariant Neural Networks". In: *J. Mach. Learn. Res.* 21 (2020), 90:1–90:61.

[2] Taco Cohen and Max Welling. In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Vol. 48. 2016, pp. 2990–2999.

[3] Marc Finzi et al. "Generalizing Convolutional Neural Networks for Equivariance to Lie Groups on Arbitrary Continuous Data". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. PMLR, 2020, pp. 3165–3176.

[4] Liyu Gong and Qiang Cheng. "Lie Group Auto-Encoder". In: *CoRR* abs/1901.09970 (2019).

[5] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[6] Michael Hutchinson et al. "LieTransformer: Equivariant self-attention for Lie Groups". In: *CoRR* abs/2012.10885 (2020).

[7] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017, pp. 5998–6008.