Equivariance and Invariance

“奇变偶不变，符号看象限”。这句经典的三角函数口诀相信能召唤起很多人的回忆，将一整张繁杂的数学结论大表浓缩为一句简单的口诀。而对于现代的深度学习技术，又有哪些“不变性”相关的故事呢。本文将主要从不变性（Invariance）与等变性（Equivariance）这两个概念的定义出发，并结合实际问题中具体的一些函数变换，探索它们在深度强化学习算法中的适用情形与处理方法。

不变性与等变性的定义

首先我们将介绍关于不变性与等变性的数学定义 [1]：

> 对于函数 $f:\mathcal{X}\to\mathcal{Y}$，$\mathcal{X}$ 和 $\mathcal{Y}$ 分别是函数的定义域与值域，即 $x\in\mathcal{X}$，$f(x)\in\mathcal{Y}$。
>
> 假如存在一个群 G，可作用于 $\mathcal{X}$ 与 $\mathcal{Y}$ 上，记为 $\star$。
>
> 可以称函数 f 关于 G 具有**不变性**，如果对于所有 $x\in\mathcal{X}$ 和所有 $g\in G$ 有：
>
> $$f(g\star x)=f(x)$$
>
> 可以称函数 f 关于 G 具有**等变性**，如果对于所有 $x\in\mathcal{X}$ 和所有 $g\in G$ 有：
>
> $$f(g\star x)=g\star f(x)$$
>
> 从定义中我们可以发现，函数相对于某个变换的不变性一般是因为函数本身是某种单射 [2]，而该变换作用于函数原像时，函数的像并不发生改变。
>
> 作为对比，函数相对于某个变换的等变性一般是因为函数本身是某种等变映射，映射前后拥有同态关系。

常见的函数变换

在这里，我们先例举一些实际决策问题中常见的函数变换关系，之后我们将讨论关于深度强化学习方法关于这些变换的不变性和等变性。

1）平移变换（Translation）

G 为 d 维平移变换算子的群 $T(d)=\{w\in\mathbb{R}^d\}$，假如向量 v 的前 k 维是位置信息：

$$w\star(v_1,...,v_n)=(v_1+w,...,v_k+w,v_{k+1},...,v_n)$$

平移变换是最常见的变换之一，一般常见于拥有空间几何性质的环境中。比如一个机器人全部活动组件的空间相对位置的向量，与其空间绝对位置的向量之间，存在该变换关系，如下图所示：

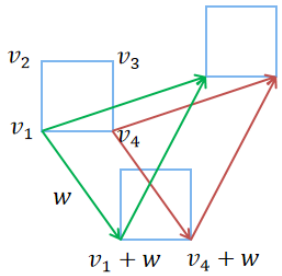

图1：平移变换示意图

2）旋转变换（Rotation）

G 为 d 维旋转变换算子的群 $SO(d)=\{Q\in\mathbb{R}^{d\times d}:Q^TQ=QQ^T=I_d,\det(Q)=1\}$

$$Q\star(v_1,...,v_n)=(Qv_1,...,Qv_n)$$

旋转变换也是一个常见的变换，不过它会比平移变换的场景略微严苛一些，存在于拥有各向同性的空间几何性质的环境中，这是因为旋转变换要求这些维度拥有相同的几何性质。比如对于一个二维平面空间内活动的智能体，如果其环境本身也存在旋转对称性，则存在该变换关系，变换矩阵为：

$$Q_{2\times2}(\theta)=\begin{pmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{pmatrix}$$

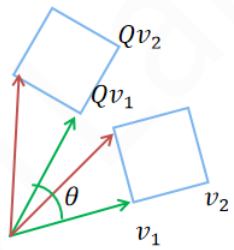

图2：旋转变换示意图

比如在 OpenDILab 的开源多智能体强化学习环境 GoBigger [3] 中，为了更好地处理多智能体的球类对象在一个开放对称的几何空间中的状态编码，就需要用到很多相关的变换的处理方式。

3）排列变换（Permutation）

G 为排列变换算子的群 $S_n=\{\sigma:[n]\to[n]\ \text{bijective function}\}$

$$\sigma\star(v_1,...,v_n)=(v_{\sigma_1},...,v_{\sigma_n})$$

排列变换常见于环境中存在多个性质相同的主体，而编码过程需要将他们的信息排列记录的场景。比如在多智能体强化学习问题中，如果多个智能体之间只有ID序号的区别，而其余属性与能力都一致，那么这意味存在编码位置交换，但实质场景依然相同的情形，此时则存在该排列变换的关系。

强化学习中的不变性与等变性

不变性与等变性的出现往往是由于某些物理定律、数学关系或环境本身的属性导致的。真实世界中存在很多的对称性，比如某些物理定律是关于空间平移或旋转不变的，总和属性与排列方式或顺序无关，等等。

因为强化学习算法是需要直接面向这些特殊的环境，建模对应的策略模型或动作价值模型。因此，我们所创建的模型，本身也应该遵循这些性质。

其中，价值函数往往符合关于某种变换的不变性，而策略函数常常符合关于某种变换的等变性。

比如对于几何对称的场景，如果环境的奖励回报也是恰好几何对称的，那么价值函数作为累计回报的期望，其数值自然与对称变换无关。而策略函数的输出是智能体的动作，当状态空间发生对称变换，为了保持环境整体的几何对称性，智能体的动作也需要进行对应的变换才能与空间一起保持一个等价状态，因此策略函数常常是关于变换等变的。

比如多智能体环境中，多个相同的智能体当前状态的价值函数，应该与这些智能体的排列无关。而此时，多智能体的策略函数则需要对排列变换具有不变性。

又比如对于诸如贪吃蛇、推箱子等二维平面环境中的智能体，其价值函数会存在关于旋转变换和反转变换的不变性，其策略函数会存在关于旋转变换和反转变换的等变性。

研究动机

如果强化学习算法的环境的属性或状态的编码中，存在着关于某些变换的不变性或等变性。我们为什么需要研究和利用它们，这主要是从两个角度去考虑的。

1）提高数据效率和训练速率

合理地利用这种不变性和等变性，可以提高强化学习算法的数据效率。

这是因为，比如我们可以使用对应的这类变换，进行数据增强，复制多种多样的新场景和数据，举一反三，创造出新的训练数据，提高数据效率。比如在训练诸如围棋的强化学习算法中，我们观测到的一种棋局，可以使用左右对称，上下对称等方式新增多种额外的棋局数据。

或者我们可以在建模中，显式地创建具有这种不变性的模型，从而让模型的输出本身天然地具备这种性质，这会客观意义上加快模型参数的训练。比如，众所周知，无跨步卷积神经网络层（CNN Layer）天然地具备一定程度的平移不变性，图神经网络（GNN）可以具有某种排列不变性。

2）提高模型鲁棒性和泛化性

从另一个角度上讲，我们可以利用这种不变性和等变性，从而创建鲁棒性和泛化能力更好的模型，减少过拟合的风险，让模型具备更好的可解释性。很多时候，强化学习算法的训练是一个从零开始学习的过程。如果利用好这种环境中的天然的规律和法则，这意味着我们可以不再完全从零开始，而是更有效地利用这些信息，将它们注入到模型中，让其具备更好的鲁棒性和泛化能力。模型的参数不再是完全随机的取值，其可训练的参数空间将被限制压缩到一个更加有效的范围内，使得强化学习智能体在训练过程中的稳定性和探索性会变得更好。

此外，环境本身所具有的这类额外信息越多，比如对于旋转对称的环境，假如对四种角度 $[\frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi]$ 构成的变换群具有不变性与等变性，那可以认为，所需决策的问题范围都可以被压缩为原来的四分之一。这样一来，神经网络学习的压力就变小了，而且在其中一种情形下学习到的“能力”可以迁移到变换群内其它任意“变换”的场景。

算法应用实例

1）在深度卷积神经网络中还原平移等变性

虽然无跨步卷积神经网络层（CNN Layer）具有平移不变性，但是因为诸如池化层、跨步卷积层等其它不具备平移等变性的网络存在，一般的深度卷积神经网络不能严格地具备平移等变性。

在论文《Making Convolutional Networks Shift-Invariant Again》[4] 中，该研究的作者设计和使用了一种抗锯齿网络（Anti-aliased network），通过引入关键的模糊池化网络（BlurPool），用模糊池化网络进行卷积计算，实现模糊池化，利用卷积计算保持平移等变性，避免了使用最大池化破坏等变性，如下图所示：

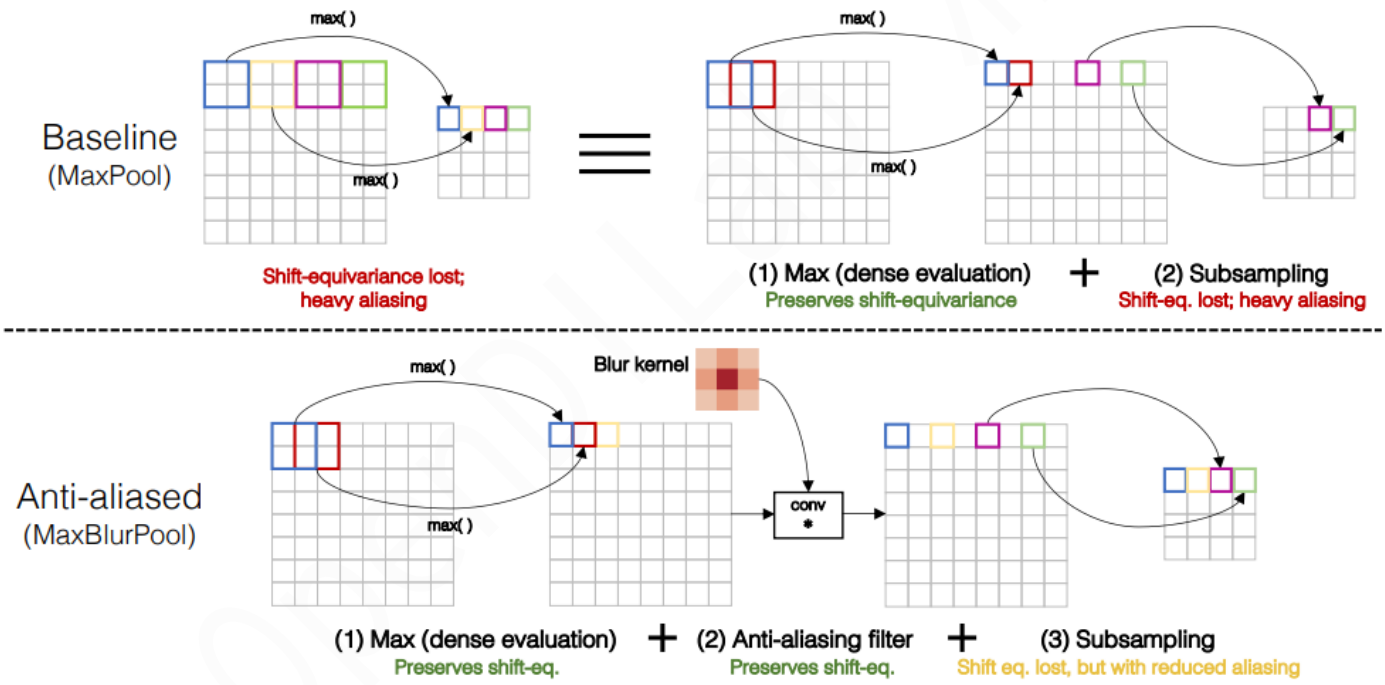

图3：普通最大池化与抗锯齿最大模糊池化的对比图

2) 在多智能体强化学习算法建模中保持对智能体输入排序的不变性和等变性

多智能体强化学习中，由于引入了多个智能体的状态和动作的编码，使得模型需要建模的动作空间和状态空间的维度和智能体的数量成比例放大，这会严重导致机器学习建模中常见的“维度灾难”现象，使得模型参数巨大，数据量需求变大，模型训练困难，继而无法成功训练至收敛。

在论文《Breaking the Curse of Dimensionality in Multiagent State Space: A Unified Agent Permutation Framework》[5] 中，该研究的作者创新性地引入了动态排列网络 (Dynamic Permutation Network, DPN) 和超策略网络 (Hyper Policy Network, HPN)，通过制造一个具有排列不

变性的网络输入层A，配合主干网络B，从而制造了拥有排列不变性的输出，如下图ABC组件所示。此外还通过两个独立的排序选择模块，将排列的等变性还原回来，从而也制造了具有排列等变性的输出，如下图ABD组件所示：

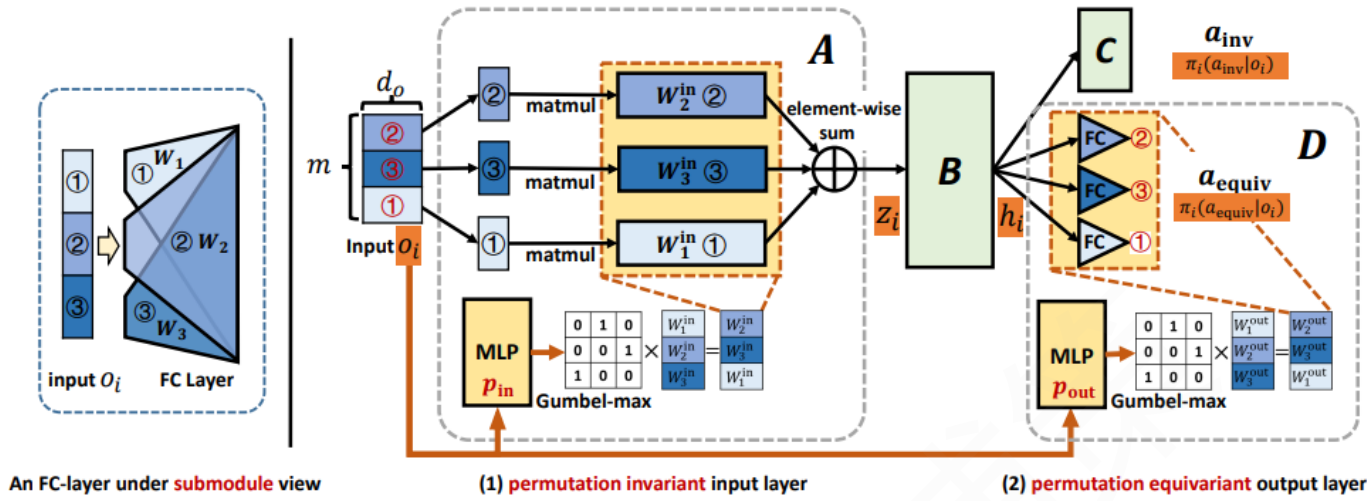

图4：使用DPN网络构建多智能体排列不变性模型输出与排列等变性模型输出的示例

3）在三维重建任务中引入旋转变换的信息

使用二维视觉输入来重建三维场景是计算机视觉领域非常重要的研究方向，是数字孪生、自动驾驶等多个应用领域的技术方案的基石。

在论文《NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis》[6] 中，研究者们发现，为了增强神经网络模型3D场景的重建能力，可以通过在模型的输入中添加自身视角的观测信息，即增加一个笛卡尔坐标系中表征视角的单位向量，从而可以让模型具有重建镜面反射效果的能力，效果十分显著，如下图所示：

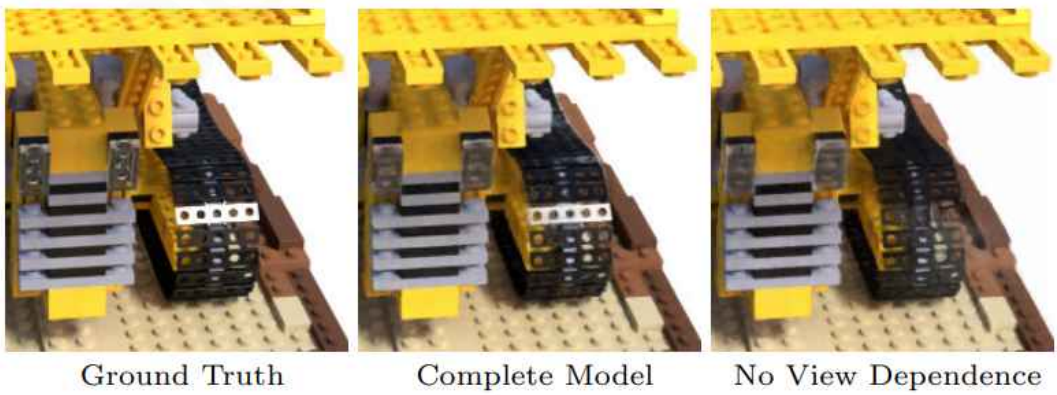

图5：3D重建中引入视角信息来还原旋转导致的镜面反射效果

开放问题和拓展思考

如何更好地利用这种不变性和等变性，在如何更为智能且高效地实现自动数据增强，如何更为合理而自然地在建模中加入这些信息，这在学术界目前依然是一个开放问题。

而从更为广义的概念上说，不变性和等变性可以看成是某种特殊的、具有对称性的约束条件。那么问题来了，对于那些更为一般普遍的，可能不具备对称性的约束条件，如果它们存在于环境和算法中，又应该如何合理有效地使用它们。这些所有问题依然在等待着作为读者们给出一个更好的答案，也欢迎和我们一起探讨。

参考文献

[1] Villar S, Hogg D W, Storey-Fisher K, et al. Scalars are universal: Equivariant machine learning, structured like classical physics[J]. Advances in Neural Information Processing Systems, 2021, 34: 28848-28863.

[2] https://en.wikipedia.org/wiki/Injective_function

[3] https://github.com/opendilab/Gobigger

[3] Zhang R. Making convolutional networks shift-invariant again[C]//International conference on machine learning. PMLR, 2019: 7324-7334.

[4] Jianye H A O, Hao X, Mao H, et al. Breaking the Curse of Dimensionality in Multiagent State Space: A Unified Agent Permutation Framework[C]//International Conference on Learning Representations.

[5] Mildenhall B, Srinivasan P P, Tancik M, et al. Nerf: Representing scenes as neural radiance fields for view synthesis[J]. Communications of the ACM, 2021, 65(1): 99-106.