PPO × Family 第三讲习题

💡 本讲习题**共包含两部分：分别是算法理论题和代码实践题**。同学们可以**选择其一项**完成并提交。（**当然也欢迎大家将两部分全部完成**，这样能够加深对课程的理解）

- 算法理论题提交方式：
  发送邮件至 opendilab@pjlab.org.cn

  请同学们严格按照下方格式命名邮箱主题/标题：

  【PPO × Family】+ 学生名 + vol.3 (第几节课）+ 作业提交日期

  示例：【PPO × Family】+ 喵小DI + vol.3 +20221207

- 代码实践题提交方式：

  PPO × Family 官方GitHub 上发起 Pull Request

  - 地址：PPOxFamily/**chapter3_obs**/hw_submission
  - PR示例：https://github.com/opendilab/PPOxFamily/pull/5
  - 命名规范：

    hw_submission(学生名称): add hw3（第几节课）+作业提交日期

    示例：hw_submission(nyz): add hw3_20230104

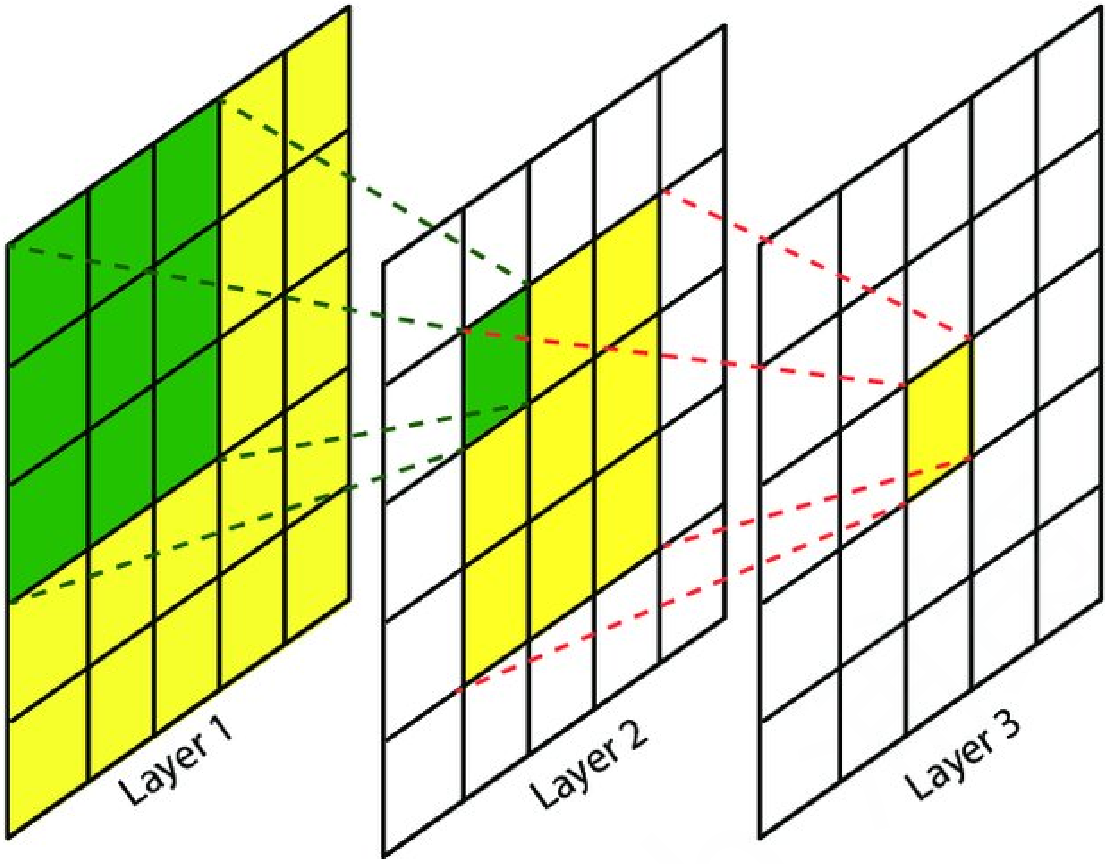
提交**截止时间为 2023.2.15 23:59 (GMT +8)**，逾期作业将不会计入证书考量。

如果其他问题请添加官方课程小助手微信（vx: OpenDILab），备注「课程」，小助手将邀请您加入官方课程微信交流群；或发送邮件至 opendilab@pjlab.org.cn

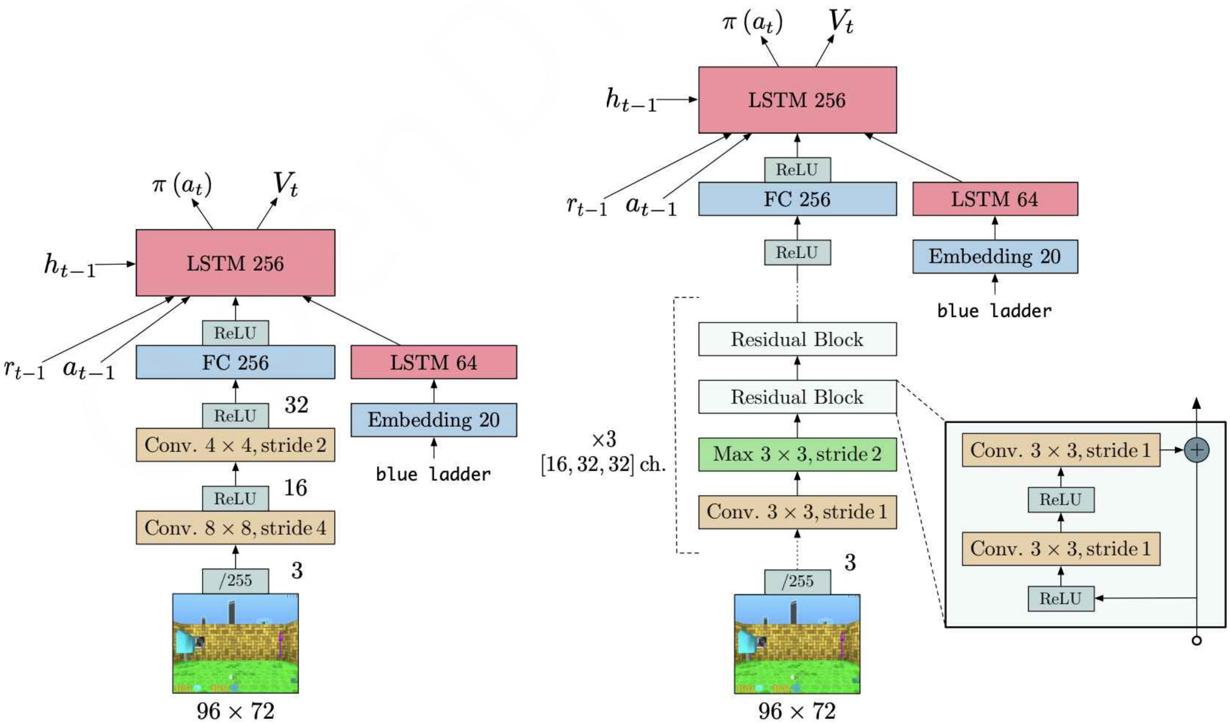
算法理论题

题目1（卷积感受野计算）

在卷积神经网络中，感受野是衡量网络结构设计的重要参考指标之一，大致可以判断某一层网络能够关联到的原始输入的大小，具体的定义为：卷积神经网络（CNN）每一层输出的特征图（feature map）上的像素点在原始图像上映射的区域大小，示例图如下（也可参阅更多关于感受野计算的资料）：



而在经典的深度强化学习 + 例如 Atari 这样的视频游戏实践中，一般常用如下类型的卷积神经网络，现在请简单计算下图中**左半部分给出的神经网络，最后一层卷积的激活值对应的感受野**，给出计算过程和最终答案。



（如有兴趣，也可以下计算右半部分，最后一个 Residual Block 的激活值对应的感受野）

题目2（表征学习网络中的 LayerNorm）

背景

Layer Normalization [1] 是一种神经网络模型中常用的归一化方法，它由多伦多大学的 Jimmy Lei Ba 等人于2016年提出（其它常用的归一化方法还包括 Batch Normalization [2] 等）。这些归一化方法的目的是将当前神经网络中激活值的数值大小缩放到一个**合适的范围**内，从而让网络的输出始终处于神经网络激活函数的有效数值范围内（例如 tanh 的线性段），并保持网络激活值的数值量级，一定程度上避免使用深层网络时常遇到的梯度爆炸或消失问题，从而让整个模型在训练中更快更好地收敛。

LN 的定义

Layer Normalization 具体实现中的定义为：假设神经网络模型某层的输出激活值为 v ，它的所有元素的均值为 μ ，方差为 σ ， v 中的元素数量为 d ：

$$\mu = \frac{1}{d}\sum_{k=1}^{d} v_k$$

$$\sigma^2 = \frac{1}{d}\sum_{k=1}^{d} (v_k - d)^2$$

那么可以将 v 归一化为符合正态分布的情况，即：

$$\bar{v} = \frac{v-\mu}{\sigma}$$

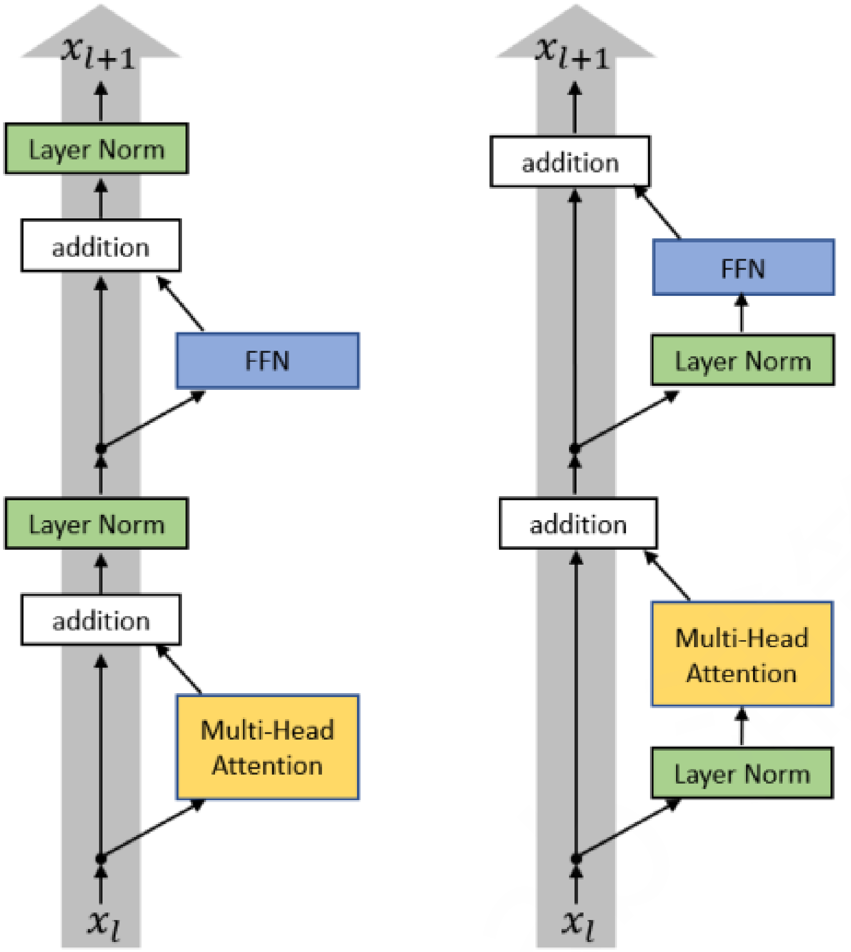
并且，将归一化后的结果进行数值缩放和偏置（仿射变换），引入缩放的尺寸参数 γ 与偏置平移的参数 β ，就可以得到 Layer Normalization 后的最终结果：

$$\mathrm{LayerNorm}(v) = \gamma\,\frac{v-\mu}{\sigma} + \beta$$

> Tips:
>
> 比如在CNN架构的网络中，某层通道数 C ，长宽为 $H \times W$ ，则共计需要对 $d = C \times \ H \times W$ 这些所有的数值做Layer Normalization。

Post-LN 与 Pre-LN

在具体的神经网络设计，例如 Transformer 结构中，有两种使用 LN 的方式（位置），即 Post-LN 和 Pre-LN，整体设计对比如下图所示（左图 Post-LN，右图 Pre-LN）：



(a) (b)

更具体地，对于 L 层 Transformer Layers 构成的模型，我们补充一些数学定义：

- 假设前馈网络的参数 W 从高斯分布 $\mathcal{N}(0, 1/d)$ 中初始化，偏置系数 b 初始化为 0 ，并将自注意力层的 $\mathrm{MultiheadAtt}$ 函数简化为 $\frac{1}{n}\sum_{j=1}^{n} x_{l,j} W^{V,l}$ ，其中 $W^{V,l}$ 从 $\mathcal{N}(0, 1/d)$ 中采样获得。
- Layer Normalization 的尺寸参数 γ 初始化为 1 ，偏置平移系数 β 初始化为 0
- 假设输入变量 $x \sim X := \mathcal{N}(0, \sigma^2 I_d)$ 也从相同高斯分布中采样获得
- 假设使用 ReLU 作为激活函数

这个 Transformer 模型在 Post-LN 和 Pre-LN 模式下完整的数据流如下表中所示：

Post-LN Transformer	Pre-LN Transformer
第 l 层，输入 x 的第 i 个元素为 $x_{l,i}^{\mathrm{post}}$	第 l 层，输入 x 的第 i 个元素为 $x_{l,i}^{\mathrm{pre}}$
$x_{l,i}^{\mathrm{post},1} = \mathrm{MultiheadAtt}(x_{l,i}^{\mathrm{post}}, [x_{l,1}^{\mathrm{post}}, \ldots, x_{l,n}^{\mathrm{post}}])$	$x_{l,i}^{\mathrm{pre},1} = \mathrm{LayerNorm}(x_{l,i}^{\mathrm{pre}})$
$x_{l,i}^{\mathrm{post},2} = x_{l,i}^{\mathrm{post}} + x_{l,i}^{\mathrm{post},1}$	$x_{l,i}^{\mathrm{pre},2} = \mathrm{MultiheadAtt}(x_{l,i}^{\mathrm{pre},1}, [x_{l,1}^{\mathrm{pre},1}, \ldots, x_{l,n}^{\mathrm{pre},1}])$
$x_{l,i}^{\mathrm{post},3} = \mathrm{LayerNorm}(x_{l,i}^{\mathrm{post},2})$	$x_{l,i}^{\mathrm{pre},3} = x_{l,i}^{\mathrm{pre}} + x_{l,i}^{\mathrm{pre},2}$

$x_{l,i}^{\text{post},4} = \text{ReLU}(x_{l,i}^{\text{post},3}W^{1,l} + b^{1,l})W^{2,l} + b^{2,l}$	$x_{l,i}^{\text{pre},4} = \text{LayerNorm}(x_{l,i}^{\text{pre},3})$
$x_{l,i}^{\text{post},5} = x_{l,i}^{\text{post},3} + x_{l,i}^{\text{post},4}$	$x_{l,i}^{\text{pre},5} = \text{ReLU}(x_{l,i}^{\text{pre},3}W^{1,l} + b^{1,l})W^{2,l} + b^{2,l}$
$x_{l,i}^{\text{post}} = \text{LayerNorm}(x_{l,i}^{\text{post},5})$	$x_{l+1,i}^{\text{pre}} = x_{l,i}^{\text{pre},5} + x_{l,i}^{\text{pre},3}$
	末层 LayerNorm： $x_{\text{Final},i}^{\text{pre}} = \text{LayerNorm}(x_{L+1,i}^{\text{pre}})$

根据上述题干信息，请证明 LN 和网络层数与每层元素数量之间的关系：

在模型网络参数初始化时，对于 Post-LN Transformer 的任意第 l 层的第二个残差网络的输出的第 i 个元素，$x_{l,i}^{\text{post},5}$，有如下公式成立：

$$\mathbb{E}(||x_{l,i}^{\text{post},5}||^2) = \frac{3}{2}d$$

在模型网络参数初始化时，对于 Pre-LN Transformer 的任意第 l 层的输入的第 i 个元素，$x_{l,i}^{\text{pre}}$，有如下公式成立：

$$(1+\frac{l}{2})d \le \mathbb{E}(||x_{l,i}^{\text{pre}}||^2) \le (1+\frac{3l}{2})d$$

提示：

- $$\mathbb{E}(||\text{ReLU}(\text{X})||^2) = \sum_{i=1}^{d}\mathbb{E}(||\text{ReLU}(\text{X}_\text{i})||^2) = \sum_{i=1}^{d}\mathbb{E}(||\text{ReLU}(\text{X}_\text{i})|X_i \ge 0||^2\mathbb{P}(X_i \ge 0)) = \frac{1}{2}d\sigma^2$$
- $\mathbb{E}(||\text{LayerNorm}(v)||^2) = \mathbb{E}(||\frac{v-\mu}{\sigma}||^2) = \mathbb{E}(\frac{\sum_{k=1}^{d}(v_k-d)^2}{\sigma^2}) = d$
- 如果你已经成功证明上述二式，就可以将其用于计算 Transformer 各层梯度的相对大小，可以进一步思考 Post-LN Transformer与Pre-LN Transformer 各层梯度的变化趋势，以及这个结论对于神经网络结构和训练方法设计的意义。

代码实践题

题目1（奇偶数预测问题）

这道题需要实现“**利用神经网络预测一个数字是奇数还是偶数**”的相关训练和测试代码。具体来说，这是一个使用神经网络进行二分类的经典问题。网络的输入是范围在 [0, 999] 内的整数，网络的输出是这个数字属于奇数还是偶数的类别标签。

这里我们提供了三种实现这个二分类问题的思路，具体代码实现可以自定义：

- 直接将需要预测的数字输入神经网络
- 先此数字转化为二进制编码，然后将所有的二进制位输入网络
- 使用三角函数 (sin, cos) 对需要预测的数字进行处理，然后输入网络

请分别实现这三种方案，提交相关代码。并回答在这三种方案中，哪些方案是可行的（即网络可以收敛）；同时思考哪种方案是实践中最优的，给出相关分析？

题目2（应用实践）

在课程第三讲（表征多模态观察空间）几个应用中任选一个

- 软体机器人（向量观察空间）
- 超级马里奥（图片观察空间）
- 羊了个羊（DI-sheep）（复杂结构化观察空间）

根据课程组给出的示例代码，训练得到相应的智能体。最终提交需要上传相关训练代码、日志截图或最终所得的智能体效果视频（replay），具体样式可以参考第三讲的示例 ISSUE。

参考文献

[1] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[2] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. PMLR, 2015: 448-456.