1. GAE：广义优势函数

首先来回顾一下优势函数，优势函数是我们在 A2C 算法中通过引入一个 baseline 得到的，其目的是表达在状态 s 下，某动作 a 相对于平均而言的优势，从数量关系来看就是随机变量相对均值的偏差，其表达式为：

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$$

GAE 借鉴了 TD 的思想，通过调整λ，得到不同的近似估计：

$$\hat{A}_t^{(1)} := \delta_t^V = -V(s_t) + r_t + \gamma V(s_{t+1}) \tag{11}$$

$$\hat{A}_t^{(2)} := \delta_t^V + \gamma \delta_{t+1}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) \tag{12}$$

$$\hat{A}_t^{(3)} := \delta_t^V + \gamma \delta_{t+1}^V + \gamma^2 \delta_{t+2}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3}) \tag{13}$$

当k无限时，这一项近似为0

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) \tag{14}$$

$$\hat{A}_t^{(\infty)} = \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \boxed{\sum_{l=0}^{\infty} \gamma^l r_{t+l}}, \tag{15}$$

the empirical returns

The generalized advantage estimator GAE(γ, λ) is defined as the exponentially-weighted average of these k-step estimators:

$$\begin{aligned}
\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} &:= (1-\lambda)\Big(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \ldots\Big) \\
&= (1-\lambda)\big(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \ldots\big) \\
&= (1-\lambda)\big(\delta_t^V(1 + \lambda + \lambda^2 + \ldots) + \gamma\delta_{t+1}^V(\lambda + \lambda^2 + \lambda^3 + \ldots) \\
&\qquad + \gamma^2\delta_{t+2}^V(\lambda^2 + \lambda^3 + \lambda^4 + \ldots) + \ldots\big) \\
&= (1-\lambda)\left(\delta_t^V\left(\frac{1}{1-\lambda}\right) + \gamma\delta_{t+1}^V\left(\frac{\lambda}{1-\lambda}\right) + \gamma^2\delta_{t+2}^V\left(\frac{\lambda^2}{1-\lambda}\right) + \ldots\right) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned} \tag{16}$$

当λ=0 和λ=1 时，情况分别为：

TD Residual近似优势函数

$$\boxed{\mathrm{GAE}(\gamma, 0): \quad \hat{A}_t := \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)} \tag{17}$$

$$\mathrm{GAE}(\gamma, 1): \quad \hat{A}_t := \sum_{l=0}^{\infty} \gamma^l \delta_{t+l} = \boxed{\sum_{l=0}^{\infty} \gamma^l r_{t+l}} - V(s_t) \tag{18}$$

the empirical returns

2. PPO 算法原理

PPO 算法之所以被提出，根本原因在于 Policy Gradient 在处理连续动作空间时 Learning rate 取值抉择困难：Learning rate 取值过小，就会导致深度强化学习收敛性较差，陷入完不成训练的局面；取值过大则导致新旧策略迭代时数据不一致，造成学习波动较大或局部震荡。且 Policy Gradient 因为在线学习的性质，进行迭代策略时原先的采样数据无法被重复利用，每次迭代都需要重新采样。

PPO 算法在迭代更新时，观察当前策略在 t 时刻智能体处于状态 s 所采取的行为概率$\pi(a_t|s_t,\theta_{new})$与之前策略所采取行为概率$\pi(a_t|s_t,\theta_{old})$，计算概率的比值来控制策略更新幅度，比值$r_t$记作：

$$r_t(\theta)=\frac{\pi(a_t|s_t,\theta_{new})}{\pi(a_t|s_t,\theta_{old})}$$

若新旧策略差异明显且优势函数较大，则适当增加更新幅度；若r_t比值越接近 1，表明新旧策略差异越小

PPO 算法课根据 **Actor 网络**的更新方式细化为含有自适应 KL-散度的 PPO-Penalty 和含有 Clipped Surrogate Objective 函数的 PPO-Clip：

（1） PPO-Penalty 基于 KL 惩罚项优化目标函数，惩罚项系数β在迭代过程中并非固定值，需要动态调整惩罚权重，其目标函数 L 可以定义为：

$$L^{KL-PEN}(\theta)=\hat{E}\left[\frac{\pi(a_t|s_t,\theta_{new})}{\pi(a_t|s_t,\theta_{old})}\hat{A}_t-\beta KL[\pi(\cdot|s_t,\theta_{old}),\pi(\cdot|s_t,\theta_{new})]\right]$$

惩罚项β的初始值选择对算法几乎无影响，原因是它能在每次迭代时依据新旧策略的 KL 散度做适宜调整，首先设置 KL 散度的阈值d_{targ}，再通过下面的表达式计算d：

$$d=\hat{E}\left[KL[\pi(\cdot|s_t,\theta_{old}),\pi(\cdot|s_t,\theta_{new})]\right]$$

如果$d<d_{targ}/1.5$，证明散度较小，需要弱化惩罚力度，β调整为$\beta/2$

如果$d > d_{targ}/1.5$，证明散度较大，需要增强惩罚力度，β调整为$\beta \times 2$

（2） PPO-Clip 直接对新旧策略比例进行一定程度的 Clip 操作, 以约束变化幅度，其目标函数 L 的计算方式为：

$$L^{CLIP}(\theta) = \widehat{E}\left[min\left(r_t(\theta)\widehat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A}_t\right)\right]$$

其中，ε表示截断系数，一般设定值为 0.2；$clip()$表示截断函数，负责限制比例r_t在区间$[1-\varepsilon, 1+\varepsilon]$之内，以保证收敛性，$L^{CLIP}$最终借助 min 函数选取为未截断与截断目标之间的更小值，形成目标下限

L^{CLIP}可以分为优势函数 A 为正数和负数两种情况：如果优势函数为正数，需要增大新旧策略比值r_t，然而当$r_t > 1 + \varepsilon$时，将不提供额外的激励；如果优势函数为负数，需要减小新旧策略比值r_t，然而当$r_t < 1 - \varepsilon$时，将不提供额外的激励，这使得新旧策略的差异被限制在合理范围内

PPO 算法的 Critic 网络更新方式与其他 Actor-Critic 类型相似，通常采用 TD error 形式；对于 Actor 网络的更新方式，PPO 可在 KL-Penalty、Clip 之间选择，经过 OpenAI 团队验证，PPO-Clip 比 PPO-Penalty 有更好的数据效率和可行性

PPO 算法流程为：

for $i = 1$ to N do
    Run policy π_θ for T timesteps, collecting $\{s_t, a_t, r_t\}$
    Estimate advantages $\hat{A}_t = \sum_{t'>t}\gamma^{t'-t}r_{t'} - V_\phi(s_t)$ Update $\pi_{old} \leftarrow \pi_\theta$
    for $j = 1$ to M do
        Select the target function L^{KLPEN} or L^{CLIP}
        Update θ by a gradient method w.r.t. L^{KLPEN} or L^{CLIP}
    end for
    for $j = 1$ to B do
        $L_{BL}(\phi) = -\sum_{t=1}^{T}\left(\sum_{t'>t}\gamma^{t'-t}r_{t'} - V_\phi(s_t)\right)^2$
        Update ϕ by a gradient method w.r.t. $L_{BL}(\phi)$
    end for
end for