# پیش گزارش آزمایش یازدهم (پیاده‌سازی الگوریتم DQN)

امیرحسین احمدی آشتیانی ۹۹۲۳۵۰۱

محمدرضا امیری ۹۹۲۶۰۴۰

محمدمهدی نوروزی ۹۹۲۳۰۸۵

## تفاوت یادگیری تقویتی با سایر روش‌ها:

### تفاوتها

1. **نوع داده مورد استفاده:**
   - یادگیری با نظارت: داده‌ها شامل ورودی‌ها و برچسب‌های خروجی مشخص هستند. مدل بر اساس ارتباط بین این ورودی‌ها و خروجی‌ها آموزش داده می‌شود.
   - یادگیری بدون نظارت: تنها ورودی‌ها در دسترس هستند، و مدل باید ساختار یا الگوهای موجود در داده را بدون برچسب کشف کند.
   - یادگیری تقویتی: از تعامل مدل با محیط استفاده می‌شود. مدل از طریق دریافت پاداش یا تنبیه آموزش می‌بیند تا استراتژی بهینه را برای دستیابی به هدف پیدا کند.
2. **تعامل با محیط:**
   - یادگیری با نظارت و بدون نظارت: داده‌های آموزشی معمولاً ثابت و از پیش آماده شده هستند.
   - یادگیری تقویتی: مدل باید به صورت فعال با محیط تعامل کند و از بازخورد (پاداش‌ها یا تنبیهات) ناشی از تصمیمات خود بیاموزد.
3. **هدف آموزش:**
   - یادگیری با نظارت: هدف، پیش‌بینی خروجی‌های دقیق برای داده‌های جدید است.
   - یادگیری بدون نظارت: کشف خوشه‌ها یا کاهش ابعاد داده است.
   - یادگیری تقویتی: یافتن استراتژی بهینه برای رسیدن به حداکثر پاداش تجمعی در طول زمان.
4. **زمان‌بندی بازخورد:**
   - یادگیری با نظارت: بازخورد (خطا) فوری و مشخص است.
   - یادگیری تقویتی: بازخورد معمولاً تأخیری است و وابسته به اقدامات زنجیره‌ای در محیط است.
5. **کاربردها:**
   - یادگیری با نظارت: طبقه‌بندی و رگرسیون، مانند تشخیص تصویر یا پیش‌بینی قیمت.
   - یادگیری بدون نظارت: خوشه‌بندی داده‌ها یا سیستم‌های پیشنهاددهنده.
   - یادگیری تقویتی: بازی‌های کامپیوتری، رباتیک، کنترل ترافیک، و حل مسائل پیچیده مانند Go و شطرنج.

## دلایل ادغام یادگیری عمیق با یادگیری تقویتی

1. **توانایی یادگیری ویژگی‌ها (Feature Learning):**
   یادگیری عمیق قادر است ویژگی‌های پیچیده و غیرخطی را از داده‌ها استخراج کند، که برای یادگیری سیاست‌های پیچیده در یادگیری تقویتی مفید است. به ویژه در مواردی که ورودی داده‌ها فرم خام مانند تصاویر یا ویدیو است.
2. **مدیریت ابعاد بزرگ حالت (State Space):**
   در یادگیری تقویتی سنتی، حل مسائل با ابعاد بزرگ و پیوسته بسیار سخت است. مدل‌های یادگیری عمیق مانند شبکه‌های کانولوشنی (CNNs) یا شبکه‌های بازگشتی (RNNs)، به کاهش این پیچیدگی کمک می‌کنند.
3. **توانایی عمومی‌سازی:**
   شبکه‌های عمیق در یافتن الگوها و عمومی‌سازی بهتر داده‌ها توانمند هستند، که در محیط‌های پویا و ناشناخته مزیت زیادی دارد.

4. **پیشرفت محاسباتی:**

پیشرفت در سخت‌افزار (مانند GPUها) و الگوریتم‌های بهینه‌سازی پیشرفته مانند Adam و RMSprop، امکان آموزش مدل‌های عمیق برای کاربردهای یادگیری تقویتی را فراهم کرده‌اند.

5. **حل مسائل بسیار پیچیده:**

یادگیری عمیق این امکان را فراهم می‌کند که یادگیری تقویتی در محیط‌هایی مانند بازی‌ها، رباتیک، و کنترل خودران‌ها موثرتر عمل کند.

## دلایل اهمیت بافر Replay:

1. **رفع مسئله همبستگی (Correlation) بین داده‌ها:**

در یادگیری تقویتی، نمونه‌های متوالی حاصل از تعامل با محیط معمولاً به شدت به یکدیگر وابسته هستند. این وابستگی می‌تواند فرآیند یادگیری را مختل کند، زیرا فرض معمول در الگوریتم‌های بهینه‌سازی این است که نمونه‌های داده مستقل و به طور یکنواخت توزیع شده‌اند.

- Replay Buffer با ذخیره تجربیات و نمونه‌برداری تصادفی از آنها، همبستگی داده‌ها را کاهش داده و فرآیند یادگیری را پایدارتر می‌کند.

2. **استفاده موثرتر از داده‌ها:**

در روش‌های کلاسیک یادگیری تقویتی، هر تجربه تنها یک بار استفاده می‌شود و پس از آن دور ریخته می‌شود. Replay Buffer امکان استفاده چندباره از تجربیات گذشته را فراهم می‌کند، که منجر به کارایی بیشتر در استفاده از داده‌های تعامل با محیط می‌شود.

3. **پایداری در فرآیند آموزش:**

یادگیری مستقیم از داده‌های متوالی می‌تواند باعث نوسانات شدید در پارامترهای شبکه عصبی شود. Replay Buffer با نمونه‌گیری تصادفی از تجربیات متنوع، این نوسانات را کاهش داده و باعث پایداری فرآیند آموزش می‌شود.

4. **آموزش یکنواخت‌تر و بهتر:**

بافر، تجربیات مختلف از زمان‌های مختلف را در خود ذخیره می‌کند، و این تنوع داده‌ها باعث می‌شود مدل به جای تمرکز بیش از حد بر تجربیات اخیر، سیاست بهینه‌تر و عمومی‌تری را بیاموزد.

**نحوه عملکرد Replay Buffer:**

- **ذخیره تجربیات:**

هر تعامل با محیط به صورت یک چهارگانه $(s, a, r, s')$ شامل وضعیت کنونی $s$ اقدام انجام‌شده $a$ پاداش دریافت‌شده $r$ و وضعیت بعدی $s'$ در بافر ذخیره می‌شود.

- **نمونه‌برداری تصادفی:**

برای به‌روزرسانی شبکه Q به جای استفاده از داده‌های متوالی، تعدادی از تجربیات به صورت تصادفی از بافر انتخاب شده و استفاده می‌شوند.

**مثال عملی:**

در بازی‌هایی مانند **Atari**، تجربه‌هایی که بازی در ابتدا از محیط می‌گیرد، حاوی رفتارهای غیرتصادفی (و گاهی ناکارآمد) هستند. Replay Buffer این تجربیات را ذخیره کرده و به تدریج تجربیات بهینه‌تر را جایگزین تجربیات قدیمی‌تر می‌کند، که منجر به یادگیری بهتر می‌شود.

## تعریف شبکه هدف:

در DQN، دو شبکه عصبی جداگانه استفاده می‌شود:

1. **شبکه اصلی (Online Network):**
   شبکه‌ای که در هر مرحله از یادگیری به‌روزرسانی می‌شود و تصمیم‌گیری‌های مدل را بر اساس $Q(s,a;\theta)$ انجام می‌دهد.
2. **شبکه هدف (Target Network):**
   نسخه‌ای از شبکه اصلی است که با وزن‌های ثابت ($\theta'$) برای مدت مشخصی ثابت نگه داشته می‌شود. این شبکه برای محاسبه مقدار هدف Q-value در به‌روزرسانی استفاده می‌شود.

**اهمیت شبکه هدف در یادگیری:**

1. **کاهش نوسانات در یادگیری:**
   اگر شبکه اصلی هم برای محاسبه مقدار هدف و هم برای پیش‌بینی مقدار Q استفاده شود، به‌روزرسانی‌های وابسته به خود باعث ناپایداری در یادگیری می‌شود. شبکه هدف با ثابت نگه‌داشتن مقادیر هدف برای مدتی، این نوسانات را کاهش می‌دهد.
2. **جلوگیری از انفجار مقادیر Q:**
   بدون شبکه هدف، مقادیر Q ممکن است به دلیل حلقه‌های بازخوردی (Feedback Loop) به شدت افزایش یافته و باعث یادگیری ناکارآمد شود.
3. **پایداری در همگرایی:**
   شبکه هدف فرآیند همگرایی به سیاست بهینه را روان‌تر می‌کند، زیرا تغییرات تدریجی در مقادیر هدف باعث تنظیم بهتر وزن‌ها می‌شود.
4. **تثبیت یادگیری در محیط‌های پویا:**
   در محیط‌هایی که پاداش‌ها یا حالات پیچیده‌اند، شبکه هدف تغییرات سریع را مدیریت کرده و به سیستم فرصت یادگیری بهتر می‌دهد.

## توضیح کامل الگوریتم Deep Deterministic Polisy Gradient یا DDPG

الگوریتم **DDPG** یک روش یادگیری تقویتی (Reinforcement Learning) در دسته‌ی **کنترل پیوسته** است که بر مبنای دو شبکه عصبی (Actor و Critic) عمل می‌کند. این الگوریتم ترکیبی از **DQN (Deep Q-Network)** و **Policy Gradient** است که برای محیط‌های با فضای عمل پیوسته طراحی شده است.

**مؤلفه‌های اصلی DDPG**

1. **شبکه Actor و Actor Target:**
   - شبکه Actor وظیفه تولید عمل (Action) را برای هر حالت (State) دارد.
   - خروجی شبکه Actor، مقدار عمل در فضای پیوسته است که با استفاده از توابع فعال‌سازی مانند Tanh محدود می‌شود.
   - شبکه Actor Target، نسخه کپی‌شده‌ای از Actor است و برای ثبات در به‌روزرسانی‌ها استفاده می‌شود.
2. **شبکه Critic و Critic Target**:
   - شبکه Critic مقدار Q-value را برای جفت (State, Action) محاسبه می‌کند.
   - هدف Critic ارزیابی کیفیت عمل پیشنهادی توسط Actor است.
   - شبکه Critic Target مشابه Actor Target، نسخه‌ای پایدار از Critic است.
3. **Replay Buffer**:

- حافظه‌ای برای ذخیره تجربیات (Transitionها) شامل (State, Action, Reward, Next State, Done).
- با نمونه‌برداری تصادفی از این حافظه، مشکلاتی نظیر همبستگی داده‌ها و تغییرات نامنظم در داده‌های ورودی به شبکه کاهش می‌یابد.

4. **نویز Ornstein-Uhlenbeck**:
   - برای ایجاد تصادفی‌بودن در انتخاب عمل و کمک به اکتشاف استفاده می‌شود.
   - این نویز به‌طور خاص برای محیط‌های با فضای عمل پیوسته طراحی شده است.