بسم الله الرحمن الرحيم

مشروع تدريب وكيل (Agent) على تعلم كيفية لعب لعبة بسيطة مثل Pong أو Flappy Bird بنفسه من خلال التجربة والخطأ.

إعداد: محمد جميل عبد المجيد الأشقر

مقدمة / إرشادات الاستخدام:

لتشغيل أمر التدريب على المشروع نكتب في نافذة Terminal الخاصة بمحرر الأكواد الخاص بنا:

```
python main.py –train
```

وسيحفظ نتائج النموذج المدرب تلقائياً كل 10.000 إطار في المسار:

```
dqn_pong_1751995584\dqn_pong_50k.pt
```

لتشغيل أمر اللعب ورؤية الوكيل يلعب نكتب في نافذ Terminal الخاصة بمحرر الأكواد الخاص بنا:

```
python main.py --play dqn_pong_1751995584/dqn_pong_50k.pt
```

لمحة عن لعبة Pong كبيئة اختبار شهيرة:

تُعَدّ لعبة Pong من أوائل ألعاب الفيديو التي استُخدمت على نطاق واسع كبيئة اختبار في مجال التعلم المعزز والذكاء الاصطناعي. اللعبة بسيطة من الناحية الميكانيكية لكنها غنية من حيث ديناميكيات التفاعل واتخاذ القرار، حيث يتحكم الوكيل في مضربٍ يتحرك عموديًا لاعتراض كرةٍ ترتد بين الجانبين. يهدف الوكيل إلى منع الكرة من تجاوز مضربه وإعادة إرسالها إلى الجهة الأخرى، مع كسب نقاط عند فشل الخصم في صد الكرة.

تتميز Pong بعدة خصائص تجعلها بيئة مثالية للأبحاث:

- بساطة القواعد: القواعد واضحة والحركات المتاحة محدودة (تحريك المضرب للأعلى أو للأسفل أو البقاء ثابتًا).
- ردود فعل فورية: يحصل الوكيل على مكافآت أو عقوبات بشكل مباشر بعد كل تبادل للكرة.
- تمثيل بصري مباشر: اللعبة تقدّم مخرجات على شكل إطارات (Frames) يمكن معالجتها كصور، مما يسهل استخدام الشبكات العصبية الالتفافية (CNN) لاستخلاص الميزات.

معيارية وتكرار التجارب: يمكن إعادة تشغيل اللعبة عددًا غير محدود من المرات مع ظروف متشابهة، مما يسمح بتكرار التجارب وضبط المعاملات.

لهذه الأسباب أصبحت Pong منذ أوائل أبحاث DeepMind نموذجًا قياسيًا لقياس أداء خوارزميات التعلم العميق بالتعلم المعزز، مثل خوارزميةDQN ، وما زالت حتى اليوم نقطة انطلاق شائعة لاختبار الأفكار الجديدة قبل الانتقال إلى بيئات أكثر تعقيدًا.

مشكلة البحث وأهداف المشروع:

تتمثل مشكلة البحث في بناء وكيل ذكي قادر على تعلّم لعب لعبة Pong اعتمادًا على مبدأ التعلّم من التجربة دون أي معرفة سابقة بقواعد اللعبة أو استراتيجيات اللعب. يتلقى الوكيل ملاحظات بيئية على شكل إطارات بصرية متتالية، ويتخذ قراراته (التحرك للأعلى أو للأسفل أو البقاء ثابتًا) بناءً على خبراته السابقة بهدف تعظيم المكافآت على المدى الطويل.

يواجه هذا التحدي البحثي عدّة صعوبات رئيسية، من أبرزها:

- عدد الإطارات الكبير المطلوب للتعلّم: إذ يحتاج الوكيل إلى معالجة مئات الآلاف من الإطارات لتجميع خبرات كافية تسمح للشبكة العصبية بالتعميم.
- الوقت الطويل اللازم للتدريب: عمليات التدريب تتطلب تكرارًا كثيفًا للحلقات والمراحل، مما يؤدي إلى استهلاك كبير للوقت الحسابي.
- حجم الذاكرة المرتفع: الاحتفاظ بتجارب اللعب السابقة في ذاكرة إعادة التشغيل (Replay Buffer) يستهلك حجمًا كبيرًا من الذاكرة العشوائيةRAM ، خصوصًا عند زيادة عدد الإطارات المخزّنة.

انطلاقًا من هذه التحديات، يهدف المشروع إلى تطوير نموذج DQN (Deep Q-Network) قادر على:

- استيعاب الإطارات البصرية المتتالية وتحويلها إلى تمثيلات مدمجة ذات دلالة.
- تحسين سياسة اتخاذ القرار (Policy) تدريجيًا عبر التفاعل المستمر مع البيئة.
- تحقيق أداء ملحوظ في لعبةPong ، مع تقليل زمن التدريب واستهلاك الموارد قدر الإمكان مقارنة بالتطبيقات التقليدية.

الدراسات السابقة والخلفية العلمية:

نبذة عن خوارزمية(Deep Q-Network)DQN :

خوارزمية DQN هي امتداد لفكرة Q-Learning الكلاسيكية، لكنها تستخدم الشبكات العصبية العميقة كدالة تقريب (Function Approximator) لتقدير قيم $Q(s, a)$ لكل حالة وإجراء. يقوم الوكيل بجمع الخبرات من خلال التفاعل مع البيئة، ثم يستخدم هذه الخبرات لتحديث النموذج العصبي بحيث يتعلم توقع المكافآت المستقبلية لكل إجراء في سياق الحالة المعطاة. تتميز DQN بقدرتها على معالجة مدخلات عالية الأبعاد مثل الإطارات البصرية في ألعاب الفيديو، وذلك بفضل استخدام الشبكات العصبية الالتفافية (CNN) لاستخراج الخصائص ذات المعنى من الصور.

ملخص عن ورقة DeepMind الأصلية (Mnih et al., 2015) :

قدمت ورقة Mnih et al., 2015 من شركة DeepMind إنجازًا بارزًا في مجال التعلم المعزز العميق، حيث أثبتت إمكانية تدريب وكيل واحد باستخدام DQN ليلعب عدة ألعاب Atari 2600 بشكل يفوق أداء البشر أحيانًا. اعتمدت الورقة على:

- ذاكرة إعادة التشغيل :(Replay Buffer) تخزين التجارب السابقة وإعادة أخذ عينات منها بشكل عشوائي لكسر الارتباط الزمني وتحقيق استقرار أكبر في التدريب.
- شبكة الهدف :(Target Network) نسخة منفصلة من الشبكة العصبية يتم تحديثها على فترات منتظمة لتجنب التذبذب السريع في القيم المستهدفة.
- المعالجة المسبقة للبيانات: تحويل الإطارات إلى صور رمادية (Gray-scale) وتقليل الأبعاد إلى 84×84 ، مع تكديس أربع إطارات متتالية لتمثيل الحركة.

نتيجة هذه الابتكارات كانت وكيلًا قادرًا على تعلّم استراتيجيات لعب متقدمة دون أي معرفة مسبقة بقواعد الألعاب، معتمدًا فقط على الملاحظات والمكافآت.

مقارنة مختصرة مع طرق أخرى:

مقارنة مع SARSA : في حين تعتمد DQN على تحديث القيم باستخدام أفضل إجراء متاح(off-policy) ، تعتمد SARSAعلى السياسة الحالية أثناء التحديث (on-policy). DQN أثبتت تفوقها في البيئات المعقدة نظرًا لقدرتها على الاستفادة من الإجراءات المثلى النظرية حتى قبل تبنيها فعليًا.

مقارنة مع Policy Gradient: خوارزميات Policy Gradient تتعلم سياسة مباشرة دون استخدام دالة قيمة، وغالبًا ما تكون فعالة في البيئات ذات مساحات أفعال مستمرة. أما DQN فتظل الخيار الأكثر شيوعًا للأفعال المنفصلة مثل Pong، وتتميز بالاستقرار وسهولة التنفيذ نسبيًا مقارنةً بالسياسات العشوائية المستمرة.

المنهجية والتقنيات المستخدمة:

يعتمد هذا المشروع على منهجية التعلّم المعزز العميق، حيث يتم تدريب وكيل (Agent) من خلال التفاعل المستمر مع بيئة لعبة .Pong يقوم الوكيل بجمع ملاحظات من البيئة على شكل إطارات بصرية متتالية، ثم يتخذ إجراءً (تحريك المضرب للأعلى أو الأسفل أو البقاء ثابتًا) بهدف تعظيم المكافأة التراكمية على المدى الطويل.

١. **البيئة**: Gymnasium + ALE (Environment) تم استخدام مكتبة Gymnasium مع واجهة ALE (Arcade Learning Environment)لتوفير بيئة Pong القياسية، والتي توفر إطارًا موحدًا للتفاعل مع اللعبة وجمع الملاحظات والمكافآت.

٢. **الأدوات البرمجية لغة البرمجة Python :نظرًا لمرونتها وتوافر مكتبات التعلم الآلي.**

مكتبات رئيسية:

PyTorch لبناء الشبكة العصبية وتدريبها.

OpenCV لمعالجة الصور (تحويل الإطارات إلى رمادية وتقليل الأبعاد).

NumPy للعمليات العددية.

Gymnasium لإدارة البيئة وجمع الخبرات.

٣. **الشبكة العصبية (Neural Network Architecture) شبكة عصبية التفافية (CNN) مكونة من:**

- ثلاث طبقات التفافية (Conv Layers) لاستخلاص الميزات من الإطارات المكدّسة.
- طبقة كاملة التوصيل (Fully Connected) لتجميع الميزات وتقدير قيم الأفعال.
- طبقة إخراج تُنتج قيمة Q(s,a) $Q(s, a)$ لكل إجراء ممكن.

٤. **آلية التدريب** Replay Buffer: ذاكرة لحفظ التجارب السابقة وإعادة أخذ عينات عشوائية منها لتقليل التحيز الزمني وتحقيق استقرار أكبر.

Target Network شبكة هدف يتم تحديثها دوريًا لزيادة استقرار التدريب.

Epsilon-Greedy استراتيجية استكشاف/استغلال، تبدأ بعشوائية عالية (ε=1.0) وتتناقص تدريجيًا حتى يتجه الوكيل إلى استغلال خبرته.

٥. **إعدادات التدريب عدد الإطارات: حتى ٥٠,٠٠٠ في التجربة الحالية (مع إمكانية زيادتها لاحقًا).**

معدل التعلم. 4− 10×1 4 − 10 × 1 :(Learning Rate)

حجم دفعة التدريب (Batch Size) :32 تجربة لكل تحديث.

تصميم النظام:

تم تصميم نظام الوكيل بحيث يتيح معالجة الإطارات البصرية من بيئة Pong وتحديث سياسة اتخاذ القرار تدريجيًا. يتكون النظام من وحدات مترابطة كما يلي:

١. **تدفق البيانات (Data Flow) :**

- مدخلات النظام: إطارات اللعبة القادمة من البيئة بمعدل ٦٠ إطارًا في الثانية تقريبًا.
- المعالجة المسبقة: يتم تحويل كل إطار إلى صورة رمادية، ثم اقتصاص الأجزاء غير المهمة (مثل شريط النقاط) وتقليص حجم الصورة إلى 84×84 84 × 84بكسل، وأخيرًا يتم تكديس أربع إطارات متتالية لتكوين حالة تعبر عن الحركة.
- الشبكة العصبية: تستقبل الإطارات المكدّسة وتُخرج قيمة Q لكل إجراء ممكن

٢. وحدات النظام:

- بيئة اللعب :(Environment) تدير تفاعل الوكيل مع اللعبة، تُعيد الملاحظات (الإطارات) والمكافآت، وتحدد متى تنتهي الحلقة.
- وكيل:DQN
- السياسة :(Policy Network) شبكة عصبية يتم تحديثها بالتدريب لتقدير قيم.Q
- شبكة الهدف :(Target Network) نسخة من الشبكة الرئيسية يتم تحديثها كل فترة لتثبيت التدريب.
- ذاكرة إعادة التشغيل :(Replay Buffer) تُخزّن خبرات (الحالة، الإجراء، المكافأة، الحالة التالية) وتعيد عينات عشوائية للتدريب.

٣. سير العمل:

- يبدأ الوكيل في البيئة ويتخذ إجراءات عشوائية في البداية (استكشاف).
- تُخزَّن الخبرات في.Replay Buffer
- عند اكتمال حجم معين من الخبرات، تبدأ عملية التدريب:
- أخذ دفعة عشوائية من الخبرات.
- حساب القيم المتوقعة والمستهدفة.
- تحديث أوزان الشبكة باستخدام خوارزمية الانتشار العكسي.(Backpropagation)
- يتم تحديث شبكة الهدف كل عدد معين من الإطارات.
- يتكرر التدريب عبر عدد كبير من الإطارات حتى تتحسن السياسة.

النتائج والتجارب:

خلال مراحل التدريب المختلفة لوكيل DQN على لعبةPong ، تم مراقبة الأداء من خلال ملاحظات المكافآت التي يحصل عليها الوكيل في كل حلقة لعب. وقد تم إجراء التجارب على مراحل متعددة بهدف تقييم تطور مهارات الوكيل مع مرور الوقت.

١. نتائج التدريب الأولية (حتى ٥٠,٠٠٠ إطار):

ي البداية، كان الوكيل يعتمد بشكل كبير على سياسة الاستكشاف العشوائي، حيث كان معدل الإبسيلون مرتفعًا (قريب من ١.٠)، مما أدى إلى أداء ضعيف نسبيًا مع مكافآت متدنية، غالبًا ما تصل إلى −٢٠ أو −٢١ نقطة لكل حلقة. كما ظهر أن الوكيل يميل إلى التحرك في أماكن ثابتة مثل أعلى الشاشة دون متابعة الكرة بشكل فعّال.

٢. التحسن التدريجي مع زيادة عدد الإطارات:

مع تقدم التدريب إلى ١٠٠,٠٠٠ إطار وأكثر، بدأ الوكيل في تحسين استراتيجيته تدريجيًا، مع ملاحظة زيادة في المكافآت المكتسبة، والتي بدأت تتراوح بين −١٥ إلى −١٠ نقاط في بعض الحلقات. يعود هذا إلى تراجع معدل الاستكشاف تدريجيًا وازدياد استغلال الوكيل للسياسات المكتسبة.

٣. حفظ النماذج المرحلية:

تم حفظ نماذج الوكيل بشكل دوري بعد كل ١٠,٠٠٠ إطار، مما أتاح مقارنة أداء الوكيل في مراحل مختلفة وإمكانية استئناف التدريب أو تشغيل اللعب باستخدام هذه النماذج.

٤.تقييم اللعب(Play Mode) :

عند استخدام النموذج المحفوظ بعد ٥٠,٠٠٠ إطار لتشغيل الوكيل في وضع اللعب، لوحظ أن الوكيل لا يزال يعاني من بعض السلوكيات غير الفعالة مثل الثبات في مكان معين، مما يشير إلى ضرورة استمرار التدريب لتحسين الأداء بشكل ملحوظ.

٥. ملاحظات عامة:

- يتطلب وكيل DQN وقتًا كبيرًا وتدريبًا مكثفًا ليصل إلى أداء جيد في ألعاب مثل Pong.
- ارتفاع عدد الإطارات وكمية الخبرات المخزنة في ذاكرة Replay Buffer يساهمان بشكل مباشر في تحسين جودة السياسة المكتسبة.
- ضبط معايير التدريب مثل معدل التعلم (Learning Rate) ومعدل استكشاف الإبسيلون (Epsilon Decay) يؤثر بشكل ملحوظ على سرعة وفعالية التعلم.

المشاكل التي واجهت المشروع والحلول:

أثناء تنفيذ مشروع وكيل DQN للعب لعبةPong ، واجهت عدة تحديات تقنية وعملية أثرت على سير العمل ونتائج التدريب، ومن أبرز هذه المشاكل:

١. استهلاك الوقت الكبير يحتاج تدريب الوكيل إلى معالجة ملايين الإطارات لتطوير سياسة لعب فعالة، مما يؤدي إلى استغراق وقت طويل جدًا، خصوصًا عند استخدام الأجهزة الشخصية ذات الموارد المحدودة. هذا التأخير يؤخر ظهور النتائج القابلة للتحليل ويزيد من تكلفة التجارب.

٢. استهلاك الذاكرة ذاكرة إعادة التشغيل (Replay Buffer) تحتفظ بعدد كبير من الخبرات السابقة لتجنب الاعتماد الزائد على العينات الحديثة فقط. هذا يؤدي إلى استهلاك كبير لذاكرة الوصول العشوائي(RAM) ، مما قد يسبب تباطؤًا في الأداء أو مشاكل في استقرار النظام، خصوصًا مع الزيادة في حجم الذاكرة المطلوبة.

٣. صعوبة في ضبط الإعدادات (Hyperparameters) تحقيق توازن مناسب بين عوامل مثل معدل التعلم، حجم الدُفعات، معدل استكشاف الإبسيلون(Epsilon Decay) ، وتردد تحديث شبكة الهدف يمثل تحديًا كبيرًا. الإعدادات غير الملائمة قد تؤدي إلى بطء التعلم، أو تجاوز التحديثات للنموذج، أو حتى فشل في التعلّم.

الحلول المقترحة:

لمواجهة هذه التحديات، تم اتخاذ الخطوات التالية:

تقليل عدد الإطارات التدريبية مؤقتًا إلى ٥٠,٠٠٠ إطار بدلاً من ٢٠٠,٠٠٠ لتقصير زمن التجربة الأولية، مع إمكانية زيادتها تدريجيًا لاحقًا.

تعديل معدل تقليل الاستكشاف (EPS_DECAY_FRAMES) ليصبح أسرع، مما يسرع انتقال الوكيل من الاستكشاف العشوائي إلى استغلال الخبرات المكتسبة.

تحسين إدارة الذاكرة عن طريق ضبط حجم Replay Buffer بما يتناسب مع الموارد المتاحة، مع مراجعة كفاءة العمليات داخل حلقة التدريب.

تجريب قيم مختلفة للمعاملات لضبط الأداء وتحقيق استقرار تدريجي في التدريب.

العمل المستقبلي:

استنادًا إلى نتائج هذا المشروع والتحديات التي تمت مواجهتها، يمكن توجيه العمل المستقبلي نحو عدة محاور لتعزيز أداء وكفاءة وكيل التعلّم المعزز:

توسيع زمن وكمية التدريب: زيادة عدد الإطارات التدريبية بشكل كبير لتوفير خبرات أكثر للوكيل، مما قد يحسن من جودة الاستراتيجية المكتسبة وأداء اللعب.

تجربة خوارزميات متقدمة: الانتقال من DQN التقليدية إلى تحسينات مثلRainbow DQN ، Double DQN، أو استخدام تقنيات التعلم المعزز القائمة على السياسات (Policy-Based RL) لتعزيز الاستقرار والتعلم الأسرع.

تحسين معالجة البيانات: استخدام تقنيات متقدمة لمعالجة الإطارات مثل التعلم الذاتي (Self-Supervised Learning)أو تقنيات ضغط البيانات لتقليل استهلاك الذاكرة والموارد الحسابية.

تطبيق النظام على بيئات أكثر تعقيدًا: توسيع نطاق التطبيق ليشمل ألعاب أو بيئات ذات ديناميكيات وسيناريوهات أكثر تعقيدًا مثل ألعاب ٣ Dأو بيئات محاكاة واقعية.

دمج التعلم المتعدد الأهداف: دمج استراتيجيات تعلم متعددة الأهداف أو استخدام تعلم متعدد الوكلاء لتعزيز قدرة الوكيل على التكيف والتعميم.

تحسين تجربة المستخدم: تطوير واجهات مرئية أفضل لعرض أداء الوكيل بشكل تفاعلي وتحليل دقيق للنتائج أثناء وبعد التدريب.