Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα

Αναφορά 4ης Εργασίας

Ομάδα 43

	Μέλος 1:	Μέλος 2:
Ον/μο:	Κορακοβούνης Δημήτριος	Αναγνωστόπουλος Θεόδωρος
Α.Μ.:	03116692	03116066

Γενικά

Το θέμα της συγκεκριμένης εργασίας ήταν η εκπαίδευση ενός δικτύου με χρήση ενισχυτικής μάθησης, ώστε να παίζει επιτυχώς σε ένα συγκεκριμένο περιβάλλον-παιχνίδι της παιχνιδομηχανής Atari.

Η ομάδα μας είναι η Ομάδα 43 και το παιχνίδι το οποίο μας ανατέθηκε είναι το Krull.

Δουλέψαμε τόσο σε περιβάλλον Kaggle αλλά και Colab για να παραλληλοποιήσουμε τις εκπαιδεύσεις των μοντέλων για μεγαλύτερη ταχύτητα.

Εκπαίδευση Αλγορίθμων χωρίς Στοχαστικότητα

Στην εργασία ξεκινήσαμε εκπαιδεύοντας έναν random Agent για να δούμε που κυμαίνονται τα σκορ τυχαίων επιλογών action ώστε να έχουμε μέτρο σύκρισης.

Υλοποιήσαμε τον τυχαίο πράκτορα και τον βάλαμε να παίξει 10 παιχνίδια. Υπολογίσαμε το μέσο σκορ του πράκτορα το οποίο και βγήκε 1514.2

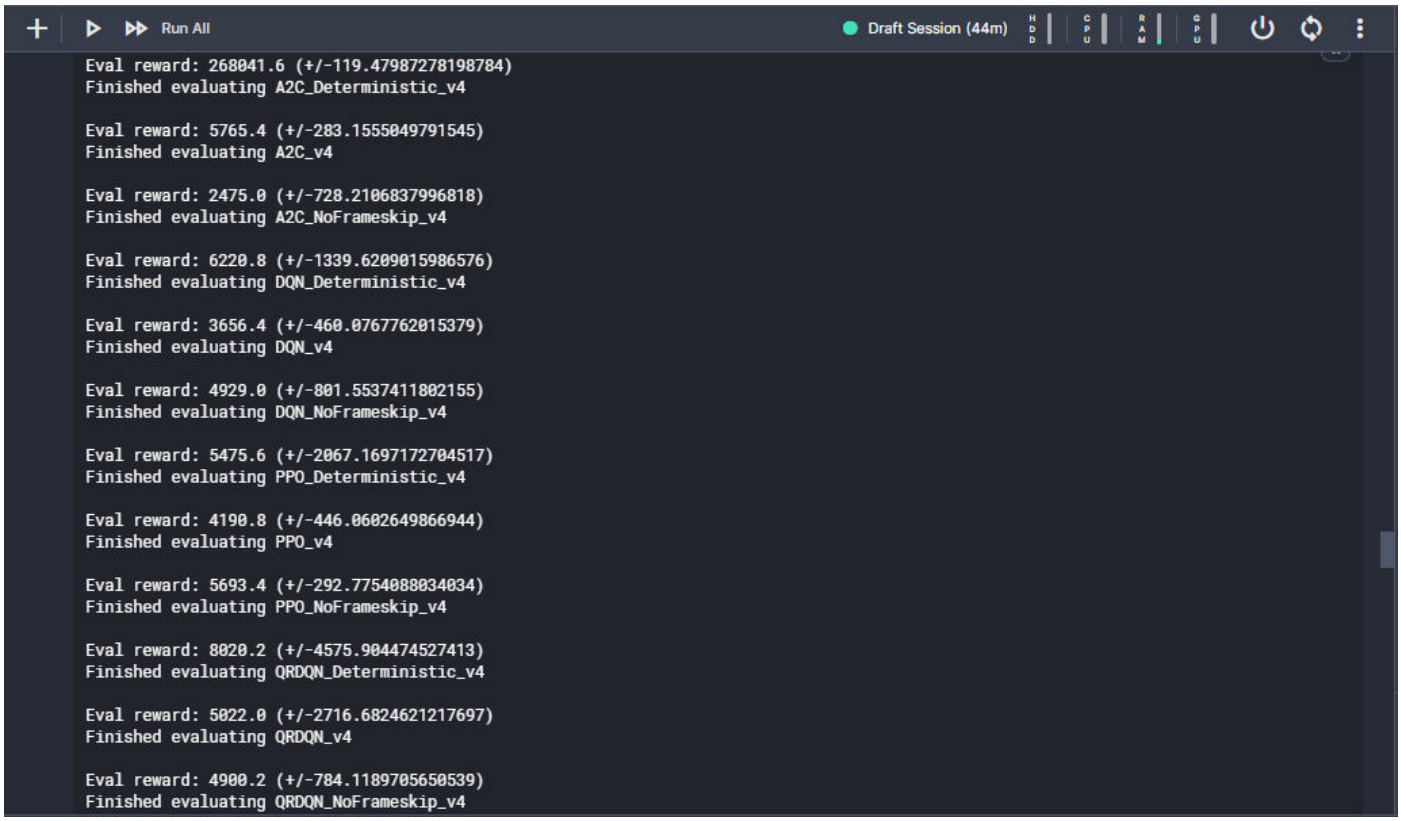
Ύστερα εκπαιδεύσαμε τα μοντέλα DQN, A2C, PPO και QRDQN με πολιτική CNN και τις default παραμέτρους.

Η χρήση CNN προτιμήθηκε καθώς θεωρήσαμε ότι θα δώσει καλύτερα αποτελέσματα δεδομένου ότι έχουμε ανάλυση εικόνας και η εξαγωγή των χαρακτηριστικών τους μπορεί να αποδειχθεί οφέλιμη.

Για την εκπαίδευση των αλγορίθμων χρησιμοποιήθηκαν 1.000.000 βήματα

Η εκπαίδευση, για μεγαλύτερη ταχύτητα έγινε σε πολλαπλά μηχανήματα και σε διαφορετικά περιβάλλοντα (όπως αναφέρθηκαμε και παραπάνω – Colab, Kaggle). Λόγω κακής διαχείρησης του χρόνου πολλά log files χάθηκαν με αποτέλεσμα η σύκριση για την επιλογή του καλύτερου αλγορίθμου να γίνει από το evaluation αυτού.

Τα αποτελέσματα της εκπαίδευσης παρατίθενται από κάτω



Από τα παραπάνω είναι ξεκάθαρο ότι ο αλγόριθμος A2C με Deterministic-v4 είχε δραματικά καλύτερα αποτελέσματα, 50-100 φορές καλύτερα από όλους τους άλλους αλγορίθμους.

Ακόμα βλέπουμε ότι έχουμε σημαντικά καλύτερα αποτελέσματα από έναν τυχαίο agent πράγμα που δείχνει ότι οι παραπάνω αλγόριθμοι αποδίδουν. Όμως δεν βλέπουμε διακριτά κάποιους αλγορίθμους ή κάποιες τεχνικές διαχείρησης περιβάλλοντος (πχ Deterministic) να ξεχωρίζουν. Αυτό μπορεί να οφείλεται στο γεγονός ότι τα 1.000.000 βήματα δεν αρκούσαν για το συγκεκριμένο παιχνίδι, ή και ακόμα στην τύχη (λιγότερο πιθανό) καθώς το αποτέλεσμα που δείχνουμε είναι το μέσο σκορ για 10 evaluations για κάθε μοντέλο.

Για τα παρακάτω βήματα θεωρούμε νικητή το Deterministic περιβάλλον, καθώς σε τρεις από τις τέσσερις περιπτώσεις (πλην του PPO) είχε καλύτερα αποτελέσματα, και το μοντέλο A2C_Deterministic_v4 και θα εργαστούμε σε αυτά.

Βελτίωση του A2C Αλγόριθμου

Στο σημείο αυτό μας ζητήθηκε να προσπαθήσουμε να βελτιώσουμε το σκορ των καλύτερων αλγορίθμων μεταβάλλοντας τις παραμέτρους. Επικεντρωθήκαμε στον A2C_Deterministic_v4 καθώς αυτός είναι που ξεχώρισε από τους άλλους με μεγάλη διαφορά.

Οι παράμετροι τις οποίες διαφοροποιήσαμε με σκοπό την βελτίωση του μοντέλου είναι οι εξής:

- Policy

Θεωρήσαμε τις 2 εκδοχές πολιτικής του δικτύου, την CNN πολοιτική και την MLP πολιτική

- Learning Rate

Θεωρήσαμε τους 2 ρυθμούς εκπαίδευσης, 0.0007 και 0.003. Η αρχική (default) τιμή είναι η 0.0007 και η τιμή που δοκιμάζουμε επιλέχθηκε με την λογική οι αλλαγές στις τιμές της πολιτικής να αλλάζουν σημαντικά όταν έχουμε μαγάλη διαφορά στο reward. Ο λόγος για τον οποίο επιλέξαμε να αυξήσουμε την τιμή είναι επειδή το παιχνίδι αποτελείται από επίπεδα, και έτσι θέλουμε η είσοδος σε επόμενο επίπεδο να έχει γρήγορη μάθηση από το exploitation αφού το exploration stage θα έχει παρέλθει.

- Discount Factor Gamma

Η συγκεκριμένη τιμή δείχνει την επιρροή που θα έχουν οι επόμενες καταστάσεις στην επιλογή της κίνησης του πράκτορα. Στην συγκεκριμένη παράμετρο η default τιμή είναι 0.99. Δώσαμε άλλη μία πιθανή τιμή για έλεγχο βελτιστοποίησης, την 0.999. Ο λόγος που την αυξήσαμε είναι ο ίδιος με τον παραπάνω, δηλαδή όταν ο agent βρεθεί στην νέα πίστα, τα reward σε αυτή να επηρεάζουν τις κινήσεις του agent.

Παρά τις βελτιστοποιήσεις που προσπαθήσαμε, δεν μπορέσαμε να βελτιώσουμε το αποτέλεσμα του αλγορίθμου. Επομένως, για την βελτιστοποίησή του, θα συνεχίσουμε απλά την εκπαίδευσή του με παραπάνω βήματα.

Συνέχιση εκπαίδευσης του A2C_Deterministic_v4 και εκπαίδευση με στοχαστικότητα (-v0)

Το αποτέλεσμα του αλγορίθμου A2C_Deterministic_v4 είναι εκπληκτικό όχι μόνο σε σύγκριση με τους αλγορίθμους που αναπτύξαμε μόνοι μας, αλλά και με τα state-of-the-art μοντέλα, τα οποία έχουνε σκορ μέχρι 23.000.

Ελπίζοντας να βελτιώσουμε τον αλγόριθμο, συνεχίσαμε την εκπαίδευση για άλλο 1 εκατομμύριο βήματα. Στις 500.000 αποθηκεύσαμε το μοντέλο και τρέξαμε τα evaluations για να δούμε εάν υπήρξε κάποια βελτίωση.

Προς έκπληξή μας, το reward είχε πέσει στο 0!

Το αποτέλεσμα αυτό έφερε μεγάλη σύγχιση, αφού ακόμη και ένας τυχαίος πράκτορας πετυχαίνει (όπως είδαμε) μεγαλύτερο σκορ.

Στο σημείο αυτό αποφασίσαμε να αφήσουμε την βελτιστοποίηση καθώς θεωρήσαμε ότι είχαμε κάνει κάποιο λάθος στην ύστερη εκπαίδευση του μοντέλου.

Εκπαιδεύσαμε, λοιπόν ένα νέο μοντέλο με στοχαστικότητα (-v0) για ένα εκατομμύριο βήματα. **Το reward και σε αυτή την περίπτωση ήταν 0!**

Αρχικά θεωρήσαμε ότι υπήρχε προγραμματιστικό λάθος. Όμως εμφανίζοντας τις κινήσεις των 2 παραπάνω πρακτόρων σε βίντεο, αντιληφθήκαμε ότι ο κώδικας είναι σωστός.

Αυτό που συνέβαινε είναι ότι ο πράκτορας εκμεταλευόταν την δυνατότητα του παιχνιδιού, να μην χάνει ποτέ.

Μηδενικό Reward – Αιτιολόγηση

Για να γίνει κατανοητό το αποτέλεσμα πρέπει να δούμε τα στάδια του παιχνιδιού και τι απαιτούν κάθε ένα από αυτά από τον πράκτορα.

Στο πρώτο στάδιο πρέπει ο agent να επιτεθεί στους εχθρούς που έρχονται να πάρουν την πριγκίπισσα. Σε αυτό το στάδιο κανένα από τα μοντέλα δεν φαίνεται να έχει κάποια συγκεκριμένη τακτική και κανένα δεν επιτίθεται σε εχθρο και δεν συλλέγει πόντους.

Στο δεύτερο στάδιο ο agent μεταφέρεται σε ένα δωμάτιο στην κάτω αριστερά γωνία. Πρέπει σε συγκεκριμένο χρονικό διάστημα να πάει στο κέντρο επάνω της οθόνης (όπου υπάρχει μία πόρτα) και ύστερα να πάει σε ένα συγκεκριμένο σημείο του τοίχου που θα υποδειχθεί μόλις φτάσει στην πόρτα. Εάν ο πράκτορας δεν πάει σε σωστό σημείο του τοίχου, τότε μεταφέρεται στο τρίτο στάδιο, όπου οφείλει να μαζέψει κάποια αντικείμενα (που δεν του δίνουν πόντους), και ύστερα μεταφέρεται ξανά στο δεύτερο στάδιο. Για να μεταφερθεί στο τέταρτο και τελευταίο πρέπει να πάει στο σωστό σημείο του δευτέρου σταδίου.Έτσι δίνεται η δυνατότητα σε κάποιο παίκτη να παίζει το παιχνίδι επ' αόριστον, μεταφερόμενος από το δεύτερο στο τρίτο στάδιο εναλλάξ.

Κατ' αυτό τον τρόπο έμαθαν και οι agents πως ακολουθώντας αυτή την τακτική δεν χάνουν ποτέ. Παρατηρώντας λοιπόν το βέλτιστο μοντέλο βλέπουμε ότι αυτό που κάνει είναι να αγνοεί την ύπαρξη της πόρτας στο κέντρο της πίστας και να μαζεύει πόντους μεχρι να φτάσει λίγο πριν το τέλος του χρόνου που του δίνεται και ύστερα να πηγαίνει τυχαία στον αριστερό τοίχο. Αυτή η κίνηση τον οδηγεί στο να μεταφερθεί πάλι στο ίδιο σημείο όπου και θα επαναλάβει την διαδικασία.Τα μοντέλα που έδιναν όμως 0 reward βλέπουμε ότι πάνε κατευθείαν στον κοντινότερο τοίχο και επαναλαμβάνουν αυτή την διαδικασία μέχρι να σταματήσει η προσωμοίωση (χωρίς να χάσουν). Η κίνηση αυτή μάλλον οφείλεται στο ότι η πολιτική του να περάσεις από το τοίχος σου δίνει περισσότερο χρόνο στο παιχνίδι, ενώ οι επόμενες κινήσεις σου δίνουν (κατά πιθανότητα) πόντους.

Για το λόγο ότι οι πράκτορες μπήκαν σε βρόχο επανάληψης που θα σταματήσει πολύ δύσκολα και αργά, δεν πραγματοποιήθηκαν περαιτέρω εκπαιδεύσεις των μοντέλων.

Ανθρώπινος παίκτης

Οι ατομικές προσπάθειες ήταν στα επίπεδα του τυχαίου πράκτορα, κοντά στα 1500. Με την χρήση όμως της τεχνικής που εκμεταλλευόντουσαν τα παραπάνω δίκτυα, φτάσαμε σε σκορ $\approx$3000 όπου και το παιχνίδι έγινε βαρετό και χάσαμε στην προσπάθεια να φτάσουμε στην επόμενη πίστα.

Actor Critic και A2C αλγόριθμος

Οι αλγόριθμοι actor critic έχουν το ιδαίτερο χαρακτηριστικό ότι χρησιμοποιούν 2 μοντέλα προς εκπαίδευση τον actor και τον critic.

Ο actor παίρνει σαν είσοδο ένα state στο οποίο βρίσκεται ο agent και υπολογίζει το action το οποίο αυτός θα πάρει.

Ο critic παίρνει σαν είσοδο το ζευγάρι (state, action) και κρίνει πόσο καλό ήταν το action που πήρε ο actor.

Κατ' αυτό το μοντέλο, στην εκπαίδευση ο critic ενημερώνει την Q (ή V) τιμή, ενώ ο actor ενημερώνει την κατανομή της πολιτικής π, σύμφωνα με την κριτική του critic.

Στην περίπτωση του A2C (Advantage Actor Critic), δεν υπολογίζεται απλά η τιμή Q για το ζευγάρι (state, action) αλλά από αυτή αφαιρείται η μέση τιμή V(state) που κερδίζεται στο συγκεκριμένο state από ένα γενικό action. Κρατάται δηλαδή η διαφορά του Q(s,a) – V(s).

Με αυτό τον τρόπο συνυπολογίζεται το πόσο καλύτερη (ή χειρότερη) είναι η επιλογή του actor από την μέση περίπτωση στο συγκεκριμένο state – εξού και το advantage.

*Σημειώνεται ότι η διαφορά Q(s,a) – V(s) δεν απαιτεί την υλοποίηση ξεχωριστού δικτύου για τον υπολογισμό του V(s), αλλά αντικαθιστούμε από την εξίσωση Bellman:

$$Q(s_t, a_t) = E[\![r_{t+1} + \gamma V(s_{t+1})]\!], \quad r_{t+1} = reword\ at\ t+1\ , \gamma = discount\ \ factor$$

Και άρα το πλεονέκτημα A(s,a) γράφεται συναρτήση του V, συνεπώς απαιτείται ένα δίκτυο για τον υπολογισμό του

Σύνδεσμος Google Drive με τα μοντέλα και το βίντεο του καλύτερου πράκτορα

https://drive.google.com/drive/folders/1bfL3NpJJsomuFUJhMPEXlWl1bPmsQLcN?usp=sharing