Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ακαδημαϊκό Έτος: 2020-2021

Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα

4η Εργαστηριακή Άσκηση

Θέμα: Παίζοντας Atari με Βαθιά Ενισχυτική Μάθηση

Τζε Χριστίνα-Ουρανία | 03116079
Ψαρουδάκης Ανδρέας | 03116001

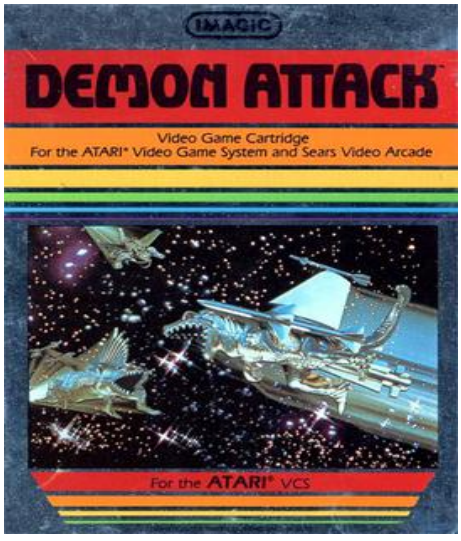
15 Μαρτίου 2021

Εισαγωγή

Σκοπός της παρούσας εργαστηριακής άσκησης είναι η εκπαίδευση αλγορίθμων βαθιάς ενισχυτικής μάθησης (deep reinforcement learning algorithms) πάνω στο βιντεοπαιχνίδι Demon Attack της Atari. Χρησιμοποιούμε τα περιβάλλοντα Atari του OpenAI gym στην εκδοχή όπου η είσοδος στο νευρωνικό είναι εικόνα, ώστε να μπορέσουν να τροφοδοτηθούν συνελικτικά δίκτυα βαθιάς μάθησης. Ωστόσο, δεν λαμβάνουμε τα περιβάλλοντα απευθείας από την OpenAI αλλά μέσα από τη βιβλιοθήκη Stable Baselines 3 (SB3). Οι αλγόριθμοι της SB3 που αξιοποιούμε είναι μόνο όσοι έχουν διακριτό χώρο ενεργειών (εξαιρούμε τον HER). Αυτοί εφαρμόζονται σε κάθε ένα από τα περιβάλλοντα που εξετάζουμε (Deterministic, "Simple" και NoFrameskip χωρίς sticky actions-v4), με σκοπό τη σύγκρισή τους ως προς τη μετρική ep_rew_mean. Στη συνέχεια, ο καλύτερος αλγόριθμος βελτιστοποιείται ως προς τις υπερπαραμέτρους του ενώ δοκιμάζεται και σε περιβάλλον με στοχαστικότητα (v0).

Λίγα λόγια για το παιχνίδι Demon Attack

Το παιχνίδι Demon Attack επινοήθηκε από τον Rob Fulop για το Atari 2600 και δημοσιεύθηκε από το Imagic το 1982. Ο παίκτης βρίσκεται στον πλανήτη Kybor και χρησιμοποιεί ένα κανόνι laser για να καταστρέψει λεγεώνες δαιμόνων που του επιτίθενται από ψηλά. Ομοίως και με άλλα διαστημικής φύσεως παιχνίδια, οι δαίμονες εμφανίζονται σε κύματα (waves) και καταλαμβάνουν την περιοχή της οθόνης έως πάνω από το κανόνι. Κάθε κύμα εισάγει νέα όπλα με τα οποία επιτίθενται οι δαίμονες, όπως λέιζερ μεγάλης ροής και συστάδες λέιζερ. Από το wave 5 και μετά, οι δαίμονες χωρίζονται σε δύο μικρότερα πλάσματα που μοιάζουν με πουλιά και τα οποία προσπαθούν να κατέβουν χαμηλά για να επιτεθούν και να νικήσουν τον παίκτη. Το παιχνίδι ήταν αρχικά προγραμματισμένο να τελειώνει μετά το 84ο κύμα, καθώς ο Fulop δεν περίμενε κανείς να φτάσει μέχρι εκεί. Δύο μόλις μέρες μετά την κυκλοφορία του, αρκετοί κατάφεραν να ολοκληρώσουν το παιχνίδι, αναγκάζοντας τον Fulop να αλλάξει μία μόνο γραμμή κώδικα ώστε να μην τελειώνει ποτέ.



Πράκτορας χωρίς μάθηση (Random Agent)

Ξεκινάμε αρχικά με έναν πράκτορα χωρίς καθόλου μάθηση. Για τον σκοπό αυτό χρειάζεται να ορίσουμε ένα περιβάλλον (test_env) και ένα μοντέλο στο οποίο θα γίνει η τελική αξιολόγηση μέσω της συνάρτησης evaluate_policy. Η διαφορά σε σχέση με την επόμενη ενότητα είναι ότι δεν πραγματοποιείται κάποιου είδους εκπαίδευση και επομένως δεν θα είχε νόημα να δοκιμάσουμε τον random agent για όλους τους συνδυασμούς περιβαλλόντων και αλγορίθμων. Ωστόσο, είναι υποχρεωτικό να ορίσουμε έναν αλγόριθμο (A2C, PPO, DQN, QR-DQN) προκειμένου να ορίσουμε το μοντέλο. Επιλέγουμε τον A2C και εξετάζουμε την μέση ανταμοιβή (score) του τυχαίου πράκτορα για όλα τα v4 περιβάλλοντα. Στον ακόλουθο πίνακα φαίνονται συγκεντρωτικά τα αποτελέσματα που λαμβάνουμε.

	Random Agent	
Environment	Reward	Variance
Deterministic-v4	69.0	±33.601
Simple-v4	200	±129.499
NoFrameskip-v4	77.0	±24.515

Παρατηρούμε ότι σε όλες τις περιπτώσεις το reward που πετυχαίνει ο random agent είναι αρκετά χαμηλό. Το γεγονός αυτό είναι αναμενόμενο αφού δεν έχει προηγηθεί κάποια εκπαίδευση.

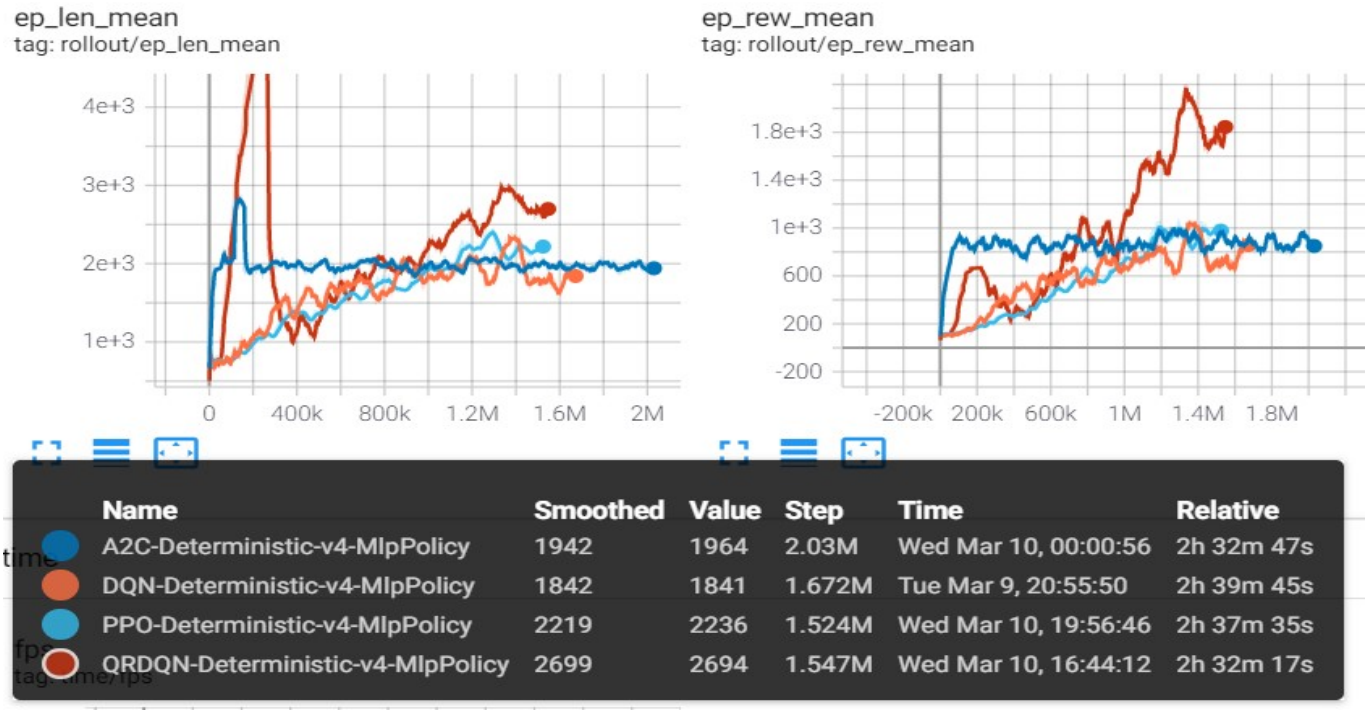
Στο επόμενο βήμα πραγματοποιείται το learning των αλγορίθμων και συνεπώς αναμένουμε μία βελτίωση των αποτελεσμάτων. Σημειώνουμε πως για λόγους περιορισμένης έκτασης της παρούσας αναφοράς οι τιμές του evaluation reward και του αντίστοιχου variance φαίνονται στα τρεγμένα κελία του υποβληθέντος αρχείου .ipynb.

Εφαρμογή αλγορίθμων ενισχυτικής μάθησης

- **Deterministic-v4**

  Αρχικά, ξεκινάμε χωρίς στοχαστικότητα και με περιβάλλον Deterministic, στο οποίο κάθε ενέργεια του πράκτορα παραμένει η ίδια για 4 frames. Δοκιμάζουμε όλους τους αλγορίθμους διακριτού χώρου ενεργειών της SB3 (πλην του HER) τόσο με Mlp όσο και με Cnn Policy.

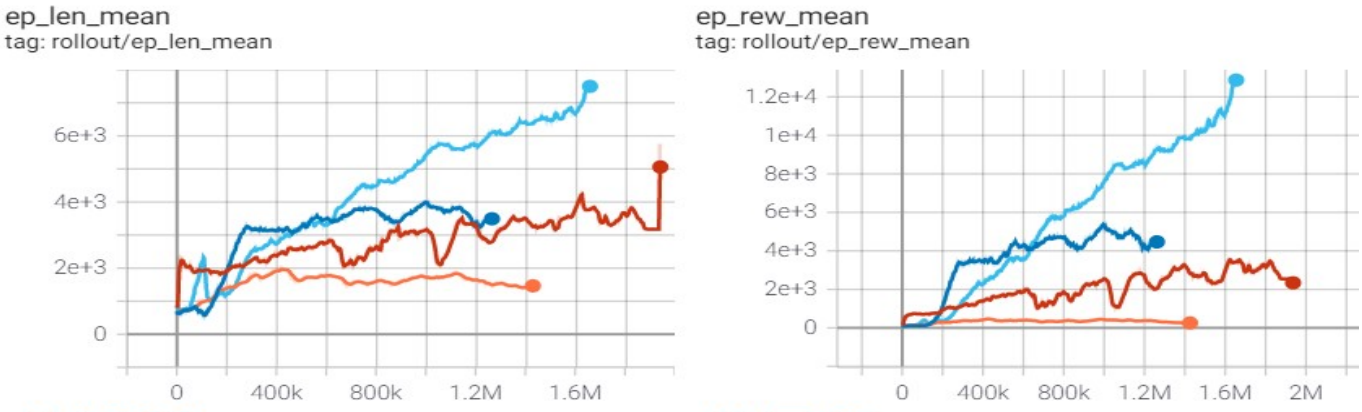
– Mlp Policy



Name	Smoothed	Value	Step	Time	Relative
A2C-Deterministic-v4-MlpPolicy	1942	1964	2.03M	Wed Mar 10, 00:00:56	2h 32m 47s
DQN-Deterministic-v4-MlpPolicy	1842	1841	1.672M	Tue Mar 9, 20:55:50	2h 39m 45s
PPO-Deterministic-v4-MlpPolicy	2219	2236	1.524M	Wed Mar 10, 19:56:46	2h 37m 35s
QRDQN-Deterministic-v4-MlpPolicy	2699	2694	1.547M	Wed Mar 10, 16:44:12	2h 32m 17s

Σχήμα 1: DemonAttackDeterministic-v4 με Mlp Policy

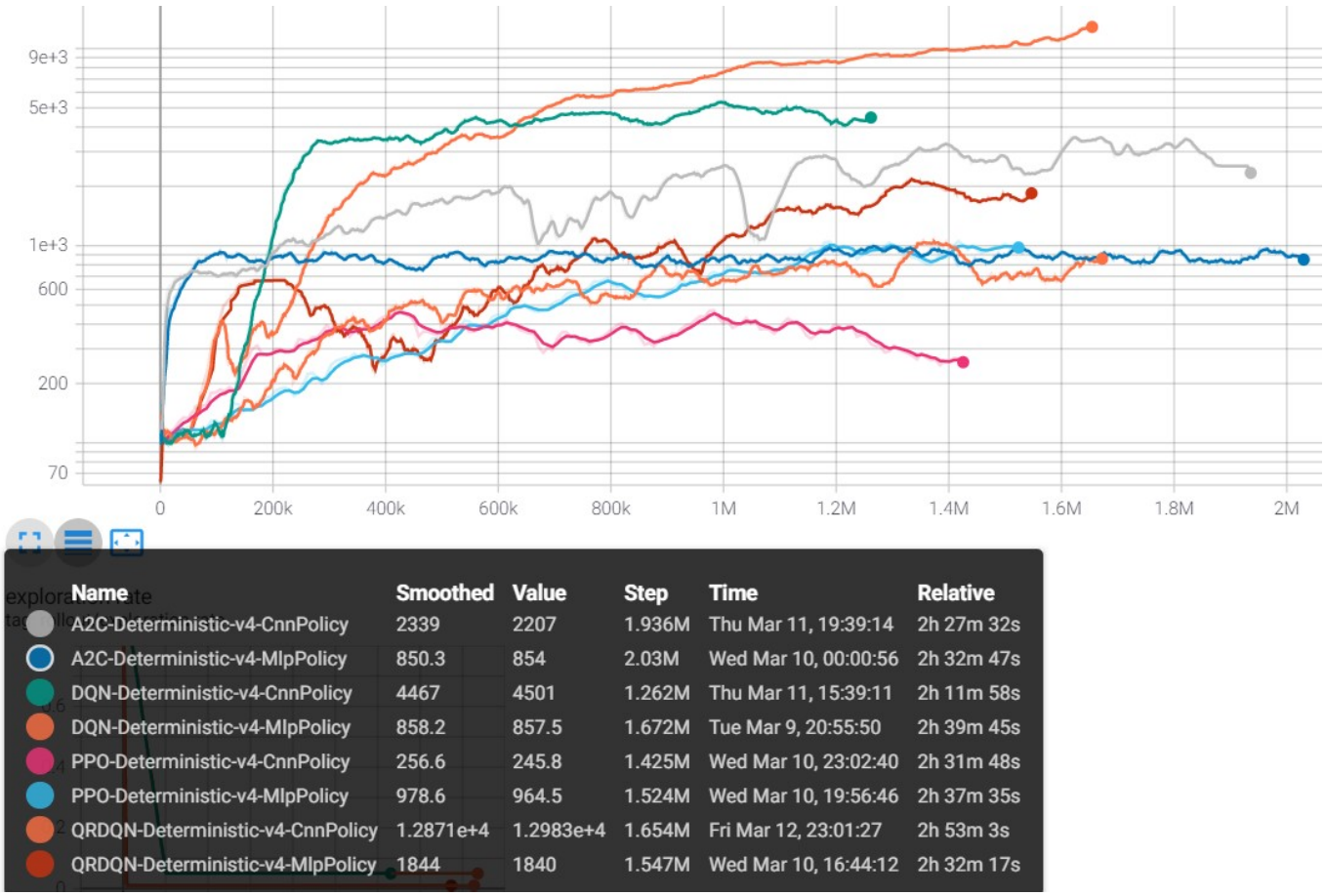
– Cnn Policy



Name	Smoothed	Value	Step	Time	Relative
A2C-Deterministic-v4-CnnPolicy	5059	5659	1.936M	Thu Mar 11, 19:39:14	2h 27m 32s
DQN-Deterministic-v4-CnnPolicy	3493	3515	1.262M	Thu Mar 11, 15:39:11	2h 11m 58s
PPO-Deterministic-v4-CnnPolicy	1465	1454	1.425M	Wed Mar 10, 23:02:40	2h 31m 48s
QRDQN-Deterministic-v4-CnnPolicy	7497	7547	1.654M	Fri Mar 12, 23:01:27	2h 53m 3s

Σχήμα 2: DemonAttackDeterministic-v4 με Cnn Policy

– Σύγκριση αλγορίθμων στο ίδιο περιβάλλον



Name	Smoothed	Value	Step	Time	Relative
A2C-Deterministic-v4-CnnPolicy	2339	2207	1.936M	Thu Mar 11, 19:39:14	2h 27m 32s
A2C-Deterministic-v4-MlpPolicy	850.3	854	2.03M	Wed Mar 10, 00:00:56	2h 32m 47s
DQN-Deterministic-v4-CnnPolicy	4467	4501	1.262M	Thu Mar 11, 15:39:11	2h 11m 58s
DQN-Deterministic-v4-MlpPolicy	858.2	857.5	1.672M	Tue Mar 9, 20:55:50	2h 39m 45s
PPO-Deterministic-v4-CnnPolicy	256.6	245.8	1.425M	Wed Mar 10, 23:02:40	2h 31m 48s
PPO-Deterministic-v4-MlpPolicy	978.6	964.5	1.524M	Wed Mar 10, 19:56:46	2h 37m 35s
QRDQN-Deterministic-v4-CnnPolicy	1.2871e+4	1.2983e+4	1.654M	Fri Mar 12, 23:01:27	2h 53m 3s
QRDQN-Deterministic-v4-MlpPolicy	1844	1840	1.547M	Wed Mar 10, 16:44:12	2h 32m 17s

Σχήμα 3: DemonAttackDeterministic-v4

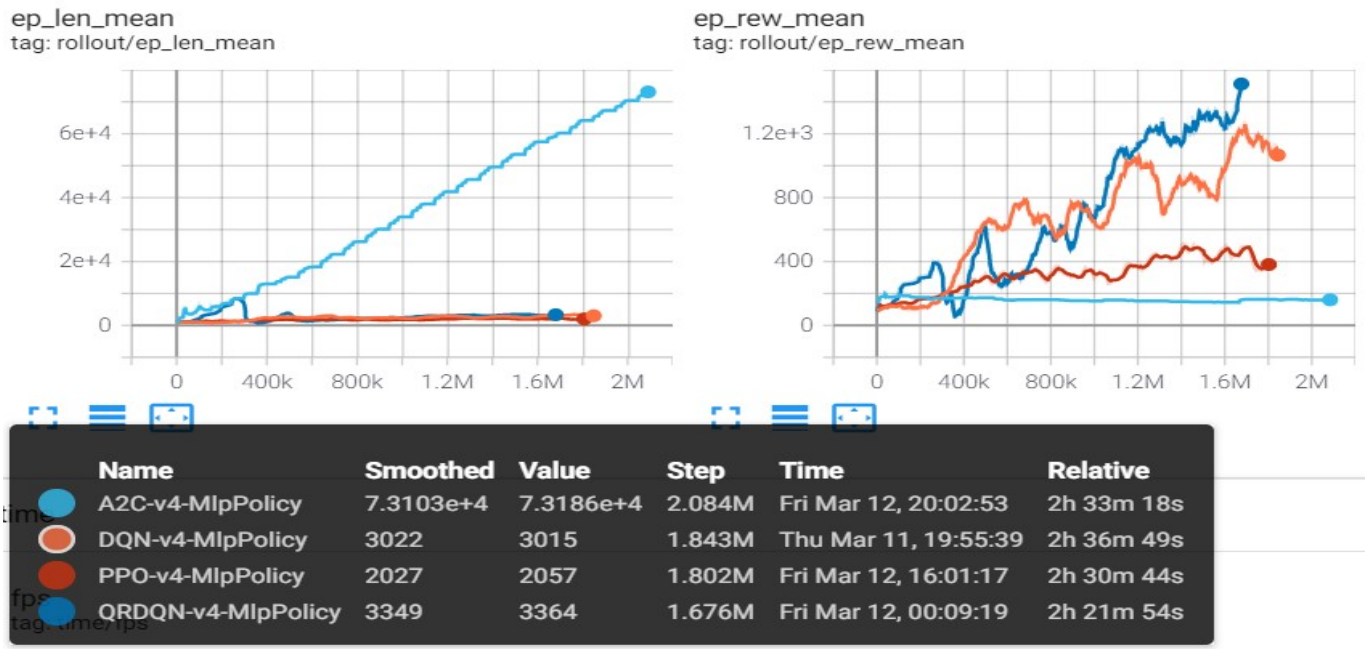
Σχολιασμός

Παρατηρούμε πως και για τα δύο policies ο QR-DQN παρουσιάζει το υψηλότερο ep_rew_mean. Ωστόσο, στην περίπτωση του Cnn, η επίδοσή του είναι σημαντικά μεγαλύτερη (μια τάξη μεγέθους). Οι υπόλοιποι τρεις αλγόριθμοι εμφανίζουν αρκετά χαμηλότερη ανταμοιβή και για τις δύο πολιτικές. Στο τελευταίο διάγραμμα φαίνεται μια συγκεντρωτική παρουσίαση της απόδοσης όλων των αλγορίθμων σε λογαριθμική κλίμακα, ώστε να υπάρχει μια καλύτερη εποπτεία και σύγκριση των επιμέρους αποτελεσμάτων. Να σημειωθεί ότι στις δύο πρώτες εικόνες οι τιμές που αναγράφονται στο υπόμνημα αφορούν το ep_len_mean και όχι το reward. Οι τιμές του ep_rew_mean είναι ορατές στο υπόμνημα της τελευταίας εικόνας. Αυτό ισχύει και για τα υπόλοιπα περιβάλλοντα ("Simple" και NoFrameskip).

- **Simple-v4**

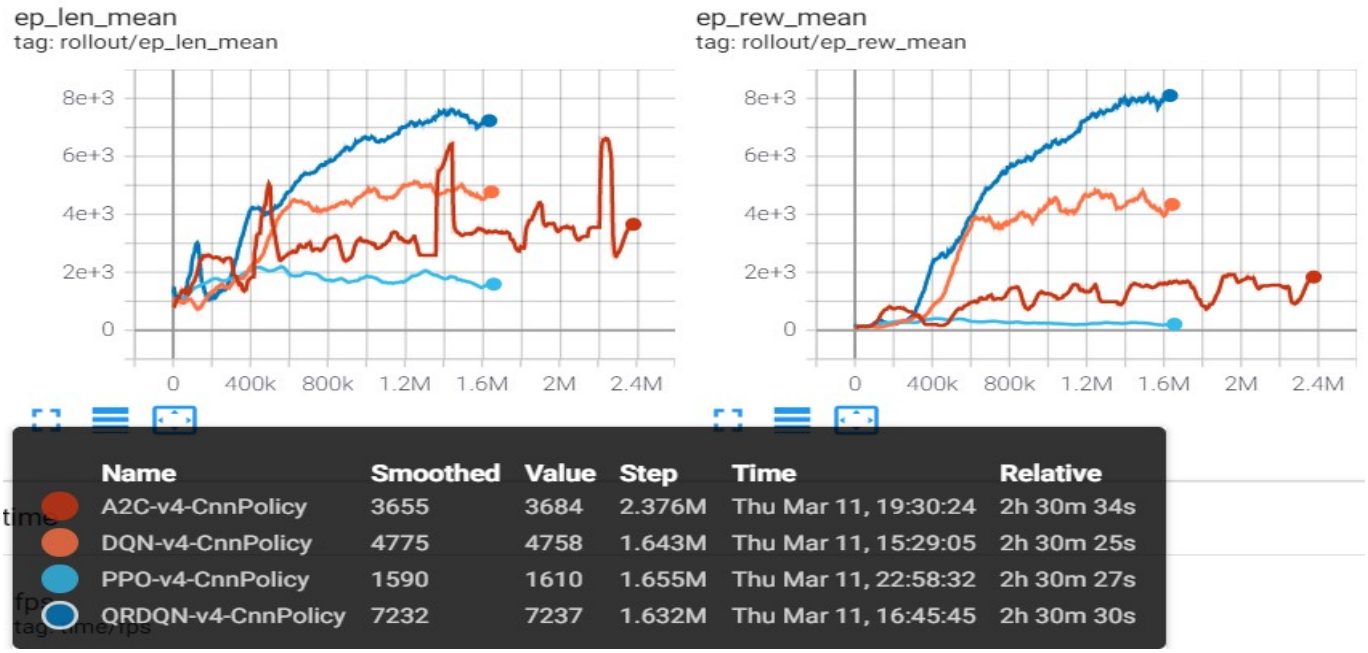
Συνεχίζουμε πάλι χωρίς sticky actions (v4) αλλά με "απλό" ("Simple") περιβάλλον, στο οποίο η διάρκεια της κάθε ενέργειας αποφασίζεται στοχαστικά και έχει διάρκεια 2-4 frames.

– Mlp Policy



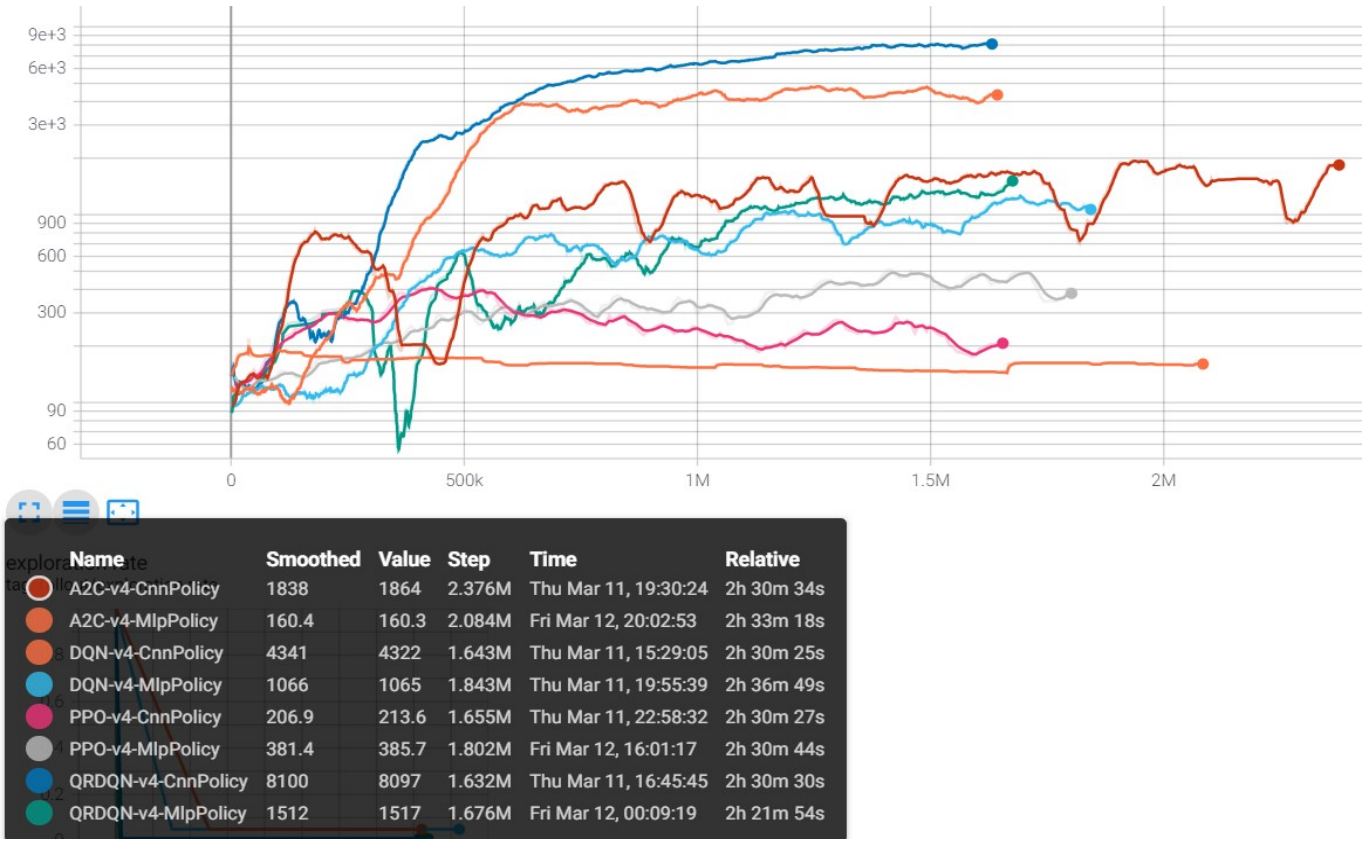
Σχήμα 4: DemonAttack-v4 με Mlp Policy

– Cnn Policy



Σχήμα 5: DemonAttack-v4 με Cnn Policy

– Σύγκριση αλγορίθμων στο ίδιο περιβάλλον



Name	Smoothed	Value	Step	Time	Relative
A2C-v4-CnnPolicy	1838	1864	2.376M	Thu Mar 11, 19:30:24	2h 30m 34s
A2C-v4-MlpPolicy	160.4	160.3	2.084M	Fri Mar 12, 20:02:53	2h 33m 18s
DQN-v4-CnnPolicy	4341	4322	1.643M	Thu Mar 11, 15:29:05	2h 30m 25s
DQN-v4-MlpPolicy	1066	1065	1.843M	Thu Mar 11, 19:55:39	2h 36m 49s
PPO-v4-CnnPolicy	206.9	213.6	1.655M	Thu Mar 11, 22:58:32	2h 30m 27s
PPO-v4-MlpPolicy	381.4	385.7	1.802M	Fri Mar 12, 16:01:17	2h 30m 44s
QRDQN-v4-CnnPolicy	8100	8097	1.632M	Thu Mar 11, 16:45:45	2h 30m 30s
QRDQN-v4-MlpPolicy	1512	1517	1.676M	Fri Mar 12, 00:09:19	2h 21m 54s

Σχήμα 6: DemonAttack-v4

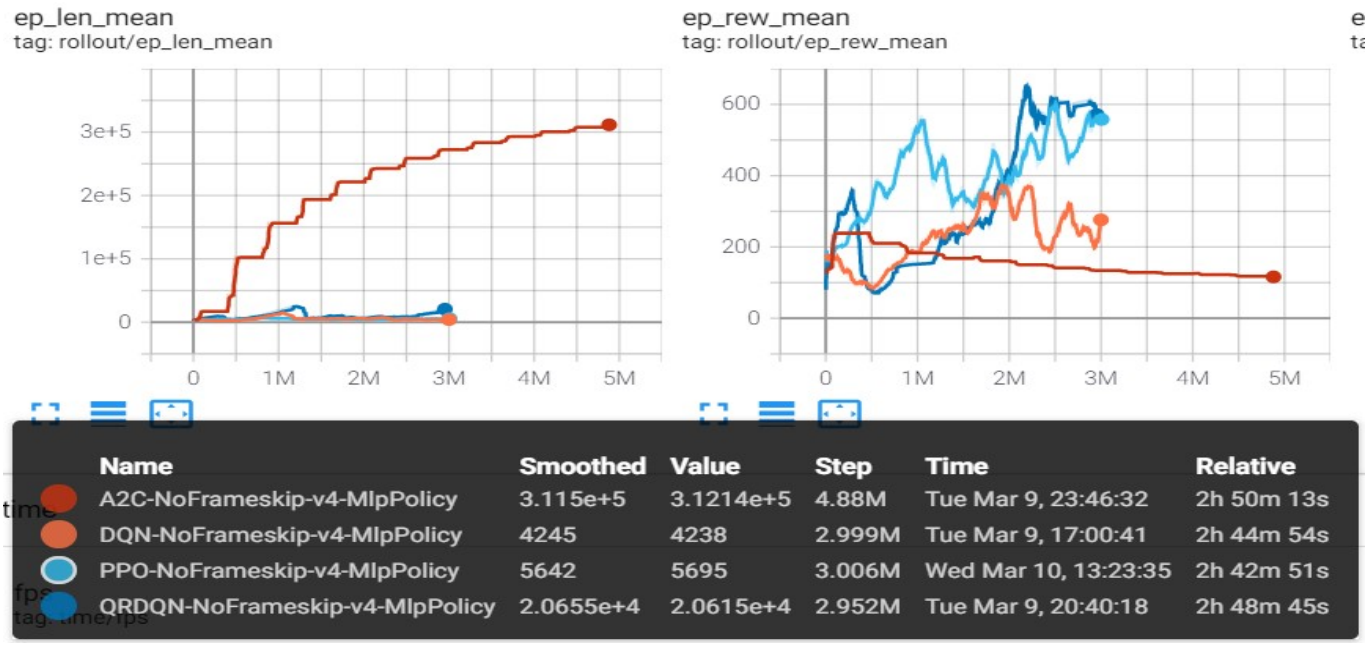
Σχολιασμός

Παρατηρούμε πως και για τα δύο policies ο QR-DQN παρουσιάζει πάλι το υψηλότερο ep_rew_mean. Επίσης, τόσο αυτός όσο και οι DQN και A2C εμφανίζουν μια σημαντική αύξηση της ανταμοιβής όταν αλλάζουμε την πολιτική από Mlp σε Cnn. Αντίθετα, ο PPO φαίνεται να εμφανίζει μια πτώση της επίδοσης κατά την μετάβαση αυτή. Σε κάθε περίπτωση πάντως το reward που πετυχαίνει είναι ιδιαίτερα χαμηλό. Αναφορικά με το χρόνο του training, όλα τα μοντέλα εκπαιδεύτηκαν για περίπου 2,5 ώρες, όπως διακρίνουμε και στο υπόμνημα. Ωστόσο, ο A2C φαίνεται να τρέχει περισσότερα timesteps στον ίδιο χρόνο, καθώς υποστηρίζει MultiProcessing.

- **NoFrameskip-v4**

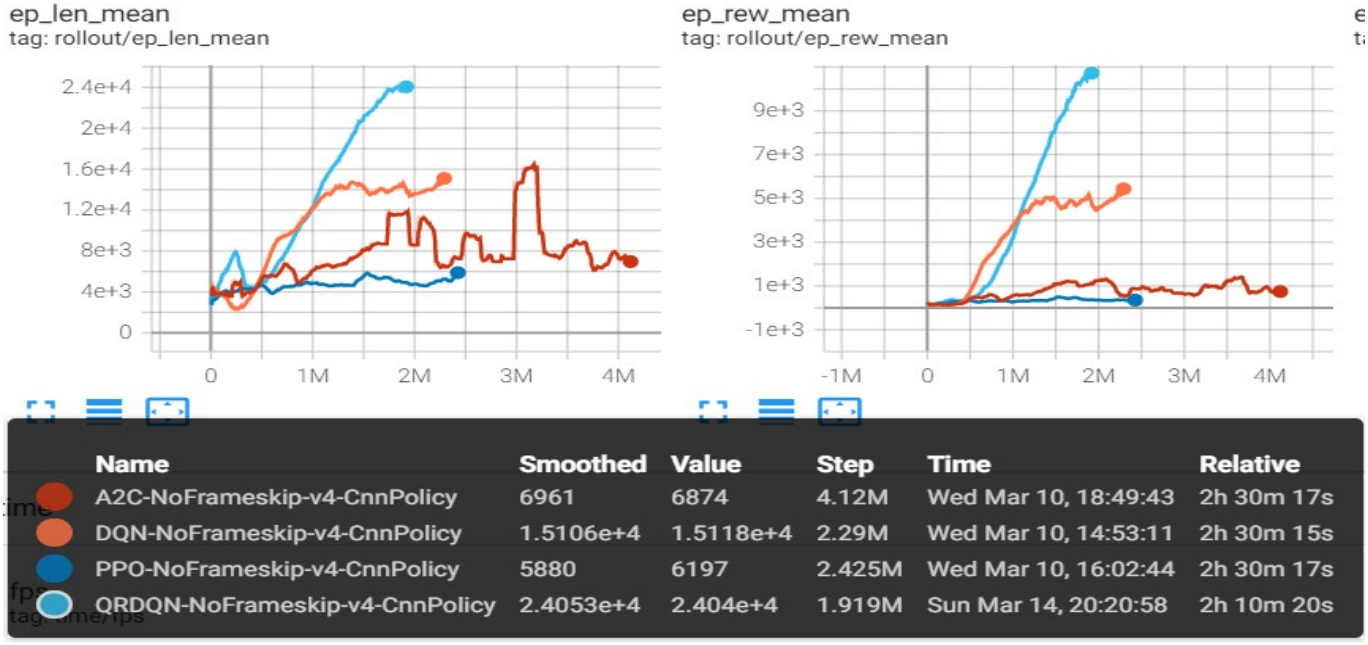
Τέλος, πειραματιζόμαστε χωρίς στοχαστικότητα και με NoFrameskip περιβάλλον, στο οποίο η ενέργεια αποφασίζεται για κάθε frame ξεχωριστά.

– Mlp Policy



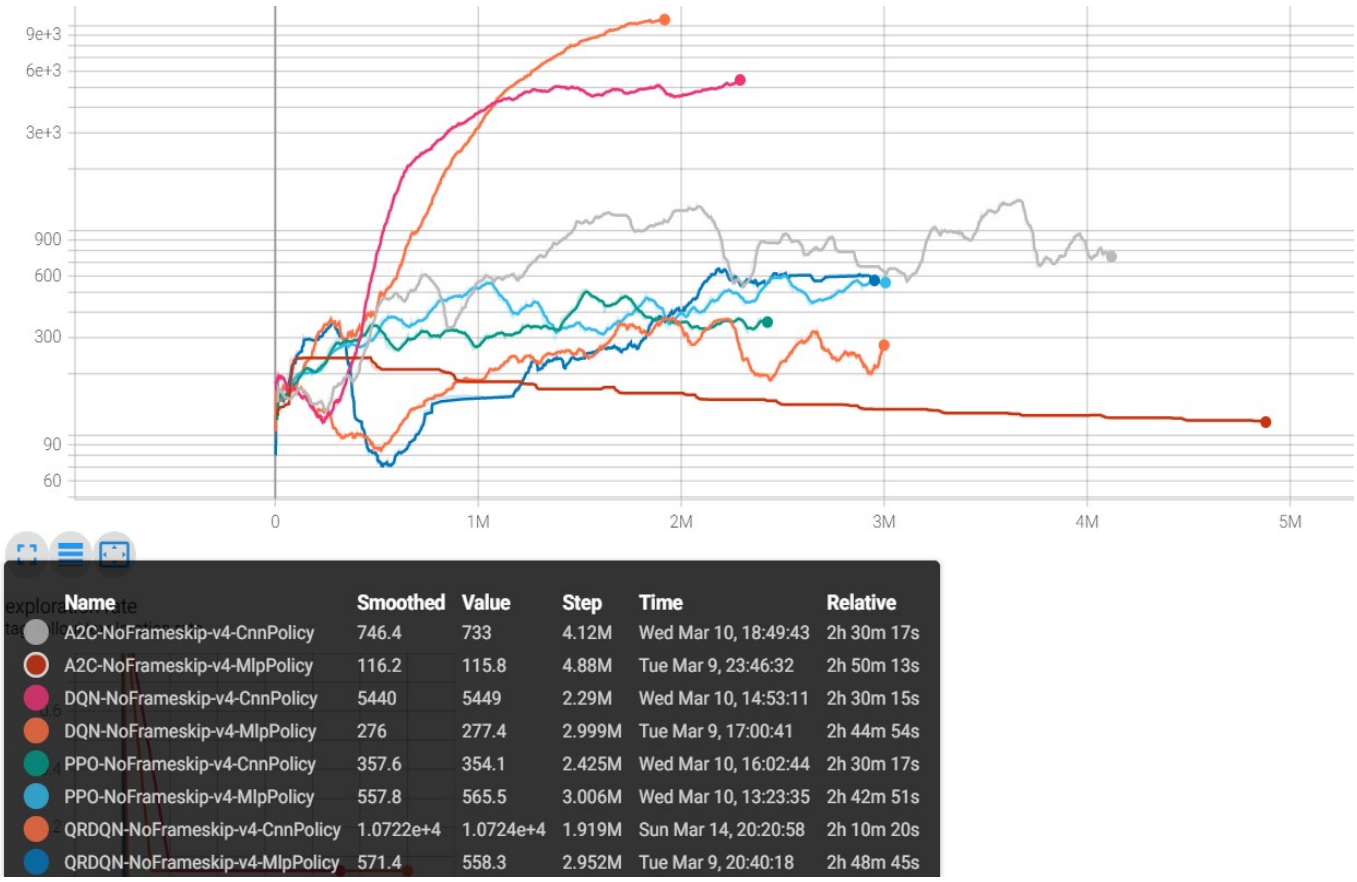
Σχήμα 7: DemonAttackNoFrameskip-v4 με Mlp Policy

– Cnn Policy



Σχήμα 8: DemonAttackNoFrameskip-v4 με Cnn Policy

– Σύγκριση αλγορίθμων στο ίδιο περιβάλλον



Name	Smoothed	Value	Step	Time	Relative
A2C-NoFrameskip-v4-CnnPolicy	746.4	733	4.12M	Wed Mar 10, 18:49:43	2h 30m 17s
A2C-NoFrameskip-v4-MlpPolicy	116.2	115.8	4.88M	Tue Mar 9, 23:46:32	2h 50m 13s
DQN-NoFrameskip-v4-CnnPolicy	5440	5449	2.29M	Wed Mar 10, 14:53:11	2h 30m 15s
DQN-NoFrameskip-v4-MlpPolicy	276	277.4	2.999M	Tue Mar 9, 17:00:41	2h 44m 54s
PPO-NoFrameskip-v4-CnnPolicy	357.6	354.1	2.425M	Wed Mar 10, 16:02:44	2h 30m 17s
PPO-NoFrameskip-v4-MlpPolicy	557.8	565.5	3.006M	Wed Mar 10, 13:23:35	2h 42m 51s
QRDQN-NoFrameskip-v4-CnnPolicy	1.0722e+4	1.0724e+4	1.919M	Sun Mar 14, 20:20:58	2h 10m 20s
QRDQN-NoFrameskip-v4-MlpPolicy	571.4	558.3	2.952M	Tue Mar 9, 20:40:18	2h 48m 45s

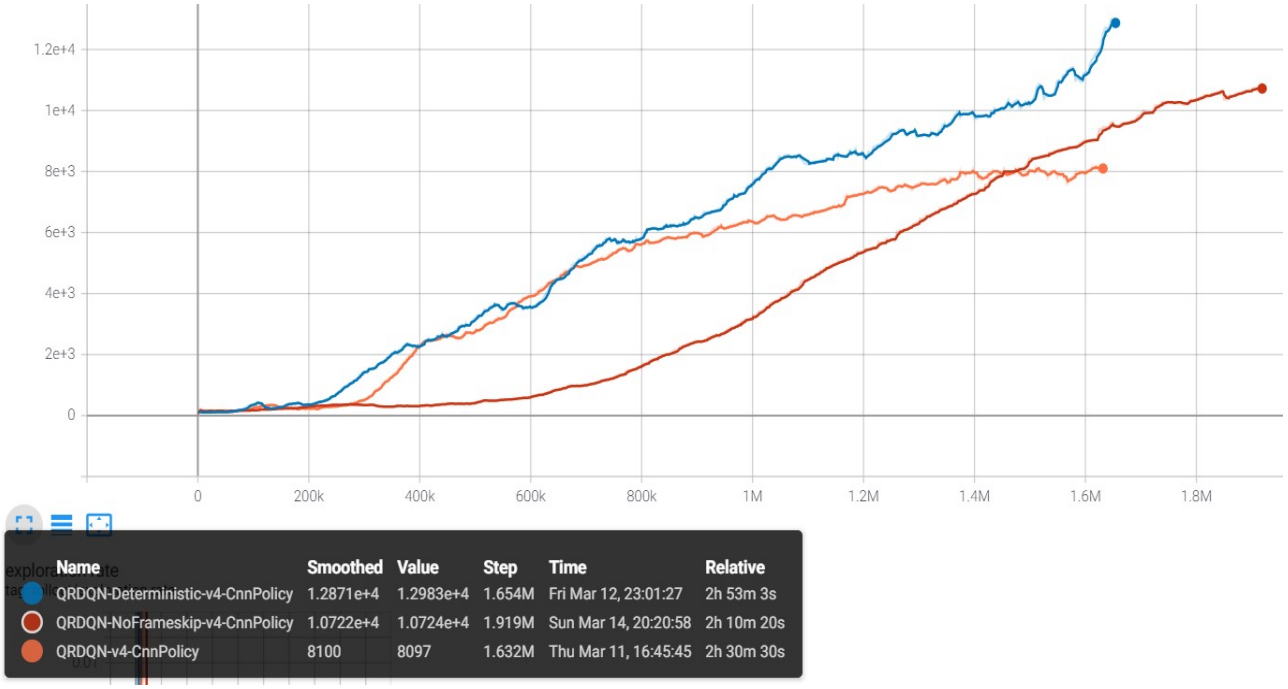
Σχήμα 9: DemonAttackNoFrameskip-v4

Σχολιασμός

Παρατηρούμε πως για την πολιτική Mlp τα rewards που σημειώνουν και οι τέσσερις αλγόριθμοι είναι πολύ χαμηλά. Ακόμα και για τους PPO, QR-DQN που παρουσιάζουν τη καλύτερη επίδοση, η ανταμοιβή τους φαίνεται να μην υπερβαίνει το 600. Αντίθετα, στην περίπτωση του Cnn Policy το ep_rew_mean αυξάνει κατά μία τάξη μεγέθους για όλους τους αλγορίθμους, με εξαίρεση τον PPO. Και εδώ, όπως και σε όλα τα προηγούμενα περιβάλλοντα που εξετάσαμε, ο QR-DQN εμφανίζει με διαφορά την υψηλότερη επίδοση.

Επιλογή βέλτιστου αλγορίθμου ανά περιβάλλον

Στο βήμα αυτό επιλέγουμε τον καλύτερο αλγόριθμο για κάθε περιβάλλον. Έπειτα, παρουσιάζουμε σε ένα κοινό διάγραμμα τις επιμέρους γραφικές παραστάσεις των επιδόσεών τους.



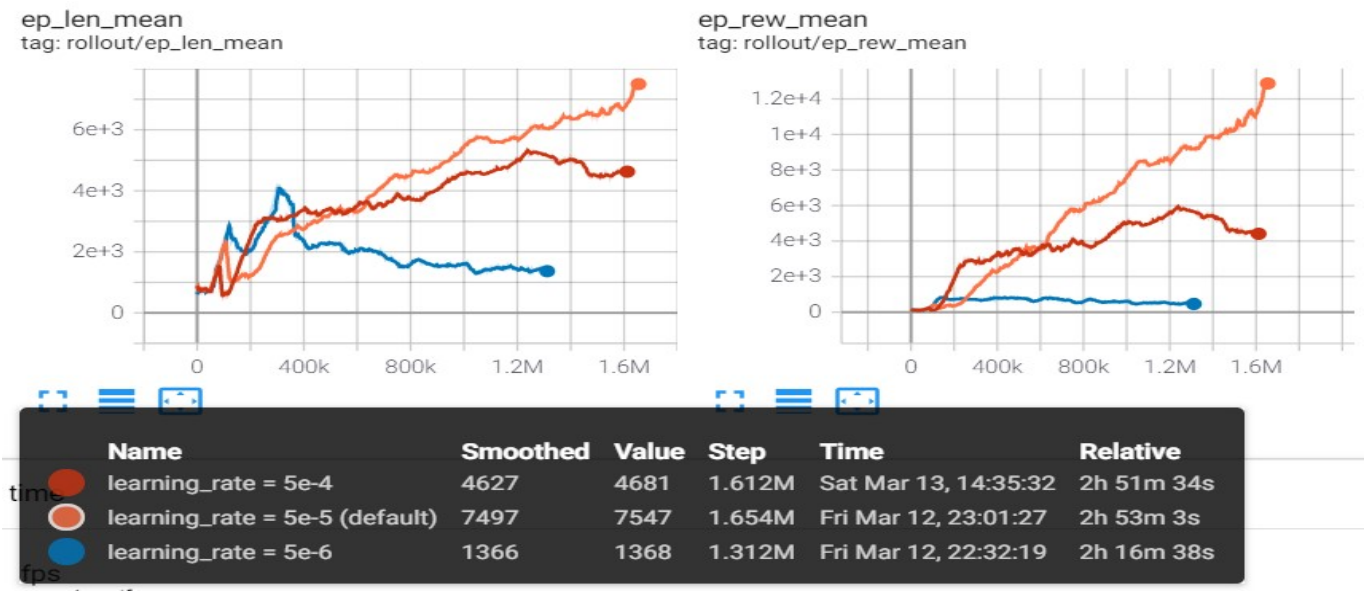
Σχήμα 10: Απεικόνιση των βέλστιστων αλγορίθμων ανά περιβάλλον

Και στα τρία περιβάλλοντα, ο βέλτιστος αλγόριθμος είναι ο QR-DQN με Cnn Policy. Ωστόσο παρατηρούμε ότι σε περιβάλλον Deterministic αυτός εμφανίζει το υψηλότερο reward και μάλιστα στον ελάχιστο δυνατό χρόνο. Συνεπώς, επιλέγουμε να βελτιστοποιήσουμε τον QR-DQN σε αυτό μόνο το περιβάλλον.

Βελτιστοποίηση με ρύθμιση υπερπαραμέτρων

Έχοντας πλέον κρατήσει το βέλτιστο συνδυασμό αλγορίθμου-περιβάλλοντος, προχωράμε σε ρύθμιση των υπεραπαραμέτρων του μοντέλου μας. Πειραματιζόμαστε με διαφορετικές τιμές του ρυθμού εκμάθησης (learning_rate), του μεγέθους δέσμης (batch size) αλλά και της παραμέτρου gamma (discount factor) με σκοπό να εξετάσουμε κατά πόσο η μεταβολή τους αυξάνει ή όχι την επίδοση (reward) του αλγορίθμου μας.

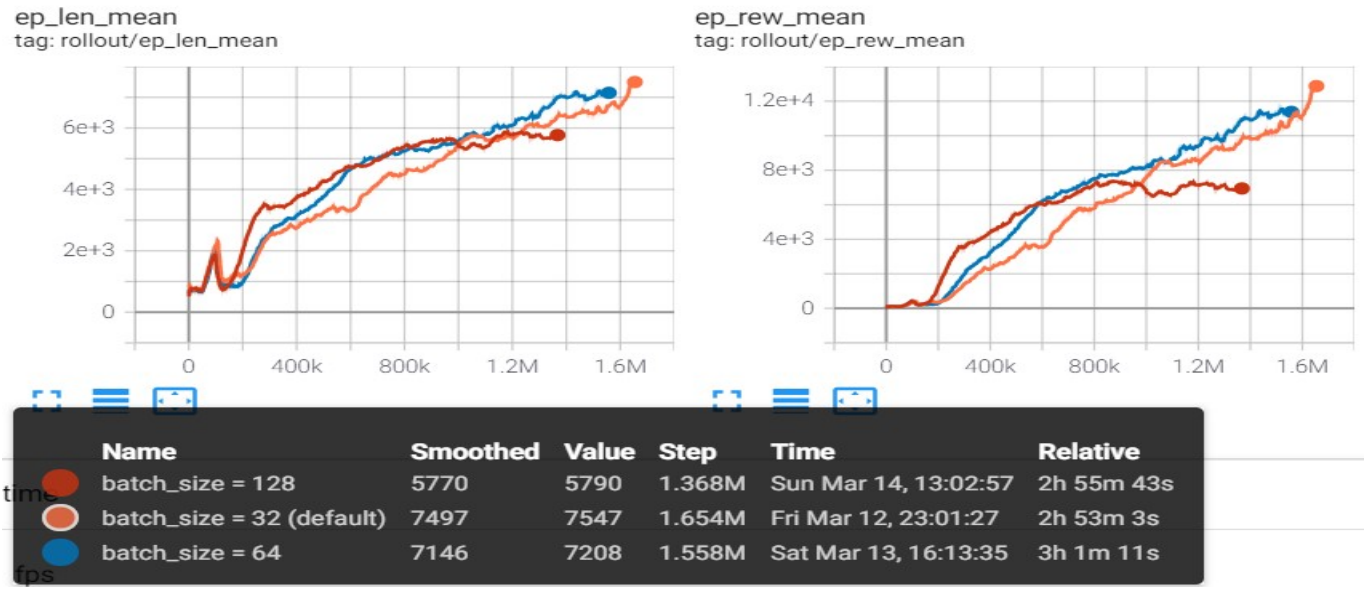
- **Learning rate**
  Ξεκινάμε μεταβάλλοντας το ρυθμό εκμάθησης (learning_rate). Σύμφωνα με το documentation η default τιμή της παραμέτρου αυτής είναι $5 \cdot 10^{-5}$. Δοκιμάζουμε τόσο να δεκαπλασιάσουμε όσο και να υποδεκαπλασιάσουμε την τιμή αυτή, διατηρώντας σταθερές όλες τις υπόλοιπες παραμέτρους του αλγορίθμου. Τα συγκεντρωτικά αποτελέσματα φαίνονται στην ακόλουθη εικόνα:



Σχήμα 11: Μεταβολή της επίδοσης για διαφορετικά learning rates

Παρατηρούμε πως η μείωση του learning_rate οδηγεί σε δραματική πτώση της απόδοσης του μοντέλου. Αντίστοιχα, και η αύξηση του ρυθμού εκμάθησης δίνει χειρότερο reward, ωστόσο αρκετά καλύτερο από αυτό της προηγούμενης περίπτωσης. Συμπαιρένουμε λοιπόν πως η default τιμή $(5 \cdot 10^{-5})$ εμφανίζει την καλύτερη δυνατή επίδοση.

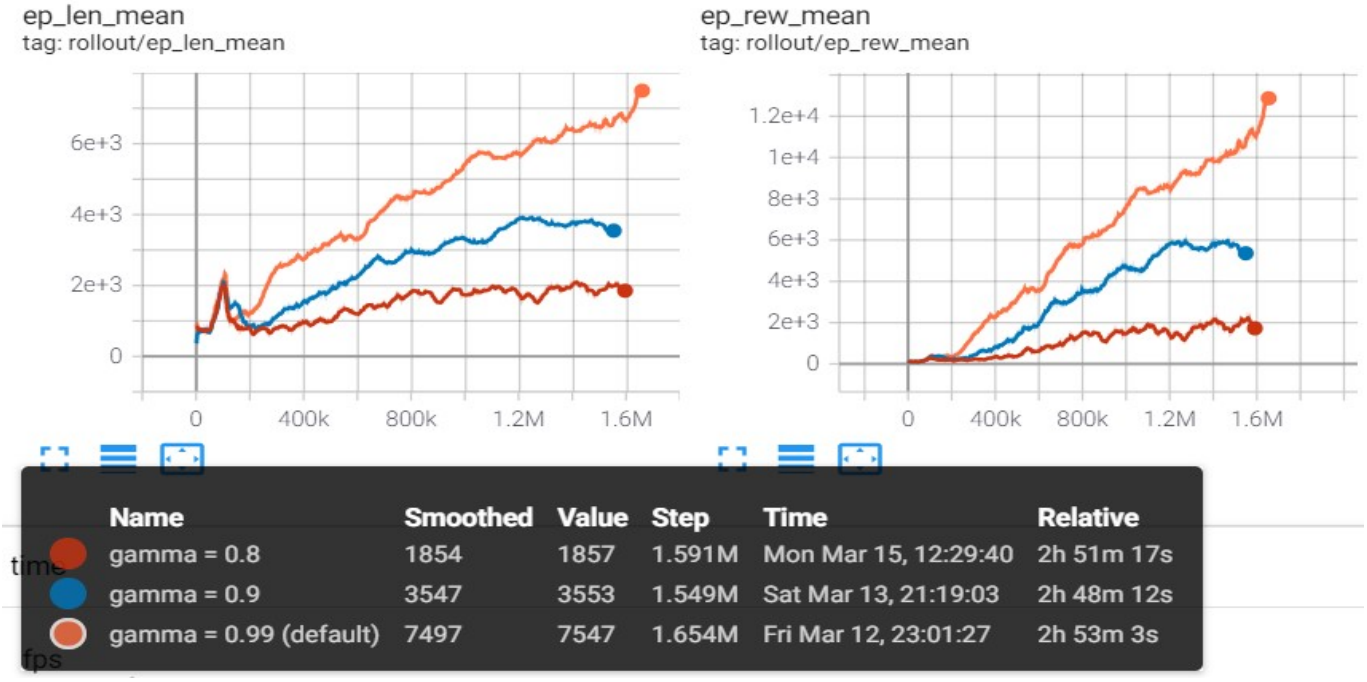
- **Batch size**
  Συνεχίζουμε δοκιμάζοντας διαφορετικές τιμές του batch size, πέραν της 32 που είναι η default. Συγκεκριμένα, αυξάνουμε διαδοχικά το μέγεθος δέσμης σε 64 και έπειτα σε 128, χωρίς όμως να μεταβάλλουμε οποιαδήποτε άλλη παράμετρο. Τα συγκεντρωτικά αποτελέσματα φαίνονται στο ακόλουθο Σχήμα:



Σχήμα 12: Μεταβολή της επίδοσης για διαφορετικά batch sizes

Ομοίως με πριν, η αύξηση του batch size φαίνεται να μην βελτιώνει το μοντέλο μας. Για μέγεθος δέσμης ίσο με 64 παρατηρείται μία συμπεριφορά παρόμοια με αυτή της default τιμής, με το reward να εμφανίζει μικρή μείωση. Σε περίπτωση περαιτέρω αύξησης από 64 σε 128 έχουμε μία σημαντική πτώση στο ep_rew_mean. Συμπεραίνουμε λοιπόν πως και εδώ η default τιμή (32) εμφανίζει την καλύτερη δυνατή επίδοση.

- **Gamma (Discount factor)**
  Τέλος, μειώνουμε την τιμή της παραμέτρου gamma από 0.99 (default) σε 0.9 και έπειτα σε 0.8. Τα συγκεντρωτικά αποτελέσματα φαίνονται στην ακόλουθη εικόνα:



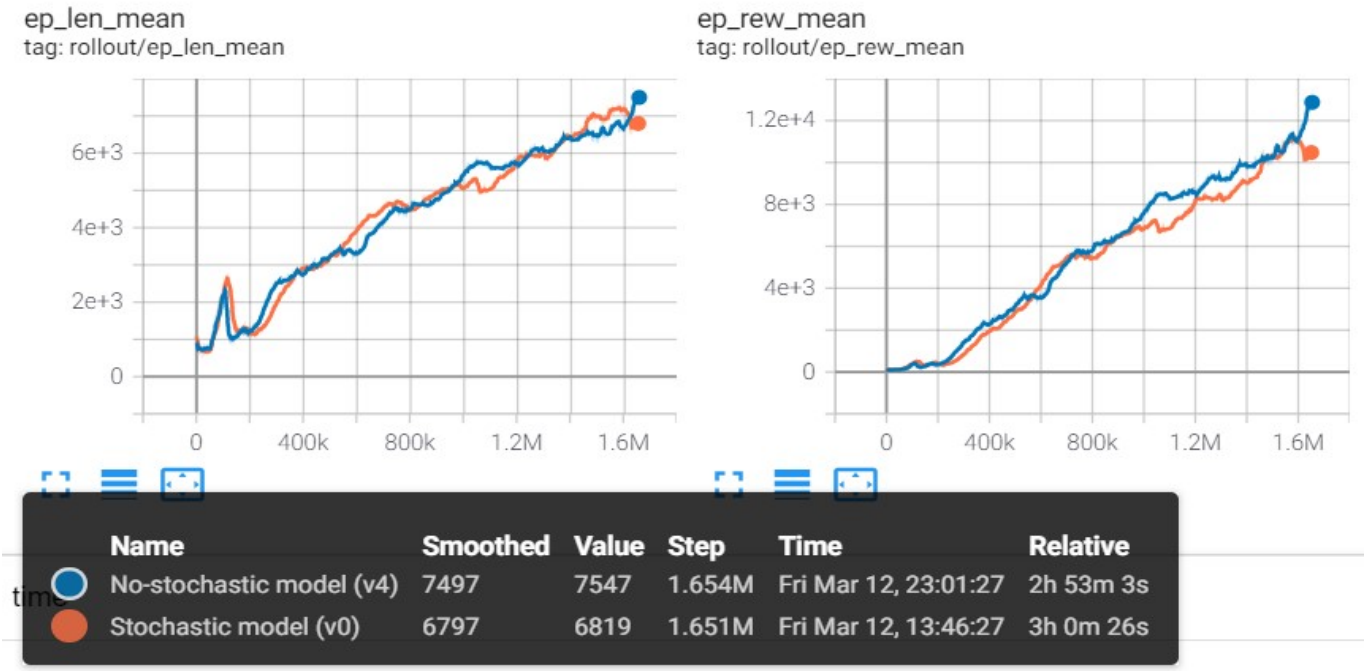
Σχήμα 13: Μεταβολή της επίδοσης για διαφορετικά discount factors

Παρατηρούμε ότι με την μείωση του discount factor σημειώνεται πτώση στην επίδοση του μοντέλου. και άρα, όπως ακριβώς και στις δύο προηγούμενες περιπτώσεις, η default τιμή δίνει το υψηλότερο reward.

Συμπερασματικά, οι default τιμές των υπερπαραμέτρων φαίνεται να βελτιστοποιούν του απόδοση του QR-DQN αλγορίθμου, γεγονός αναμενόμενο καθώς αυτές είναι ήδη optimized για Atari Games.

Δοκιμή στοχαστικότητας

Έχοντας πλέον καταλήξει στο βέλτιστο μοντέλο (QR-DQN αλγόριθμος σε Deterministic περιβάλλον) και τις βέλτιστες τιμές των υπερπαραμέτρων (οι οποίες τελικά συμπίπτουν με τις προκαθορισμένες) δοκιμάζουμε να εφαρμόσουμε στοχαστικότητα. Στην περίπτωση αυτή, ο πράκτορας στην επόμενη κίνηση δεν θα κάνει απαραίτητα αυτό που έχει αποφασίσει αλλά με μια μικρή πιθανότητα p θα κάνει την προηγούμενη ενέργειά του (κολλημένη ενέργεια, sticky action). Στο ακόλουθο Σχήμα φαίνεται η σύγκριση της επίδοσης του βέλτιστου αλγορίθμου για πιθανότητες sticky action ίσες με 0 (v4) και 0.25 (v0).

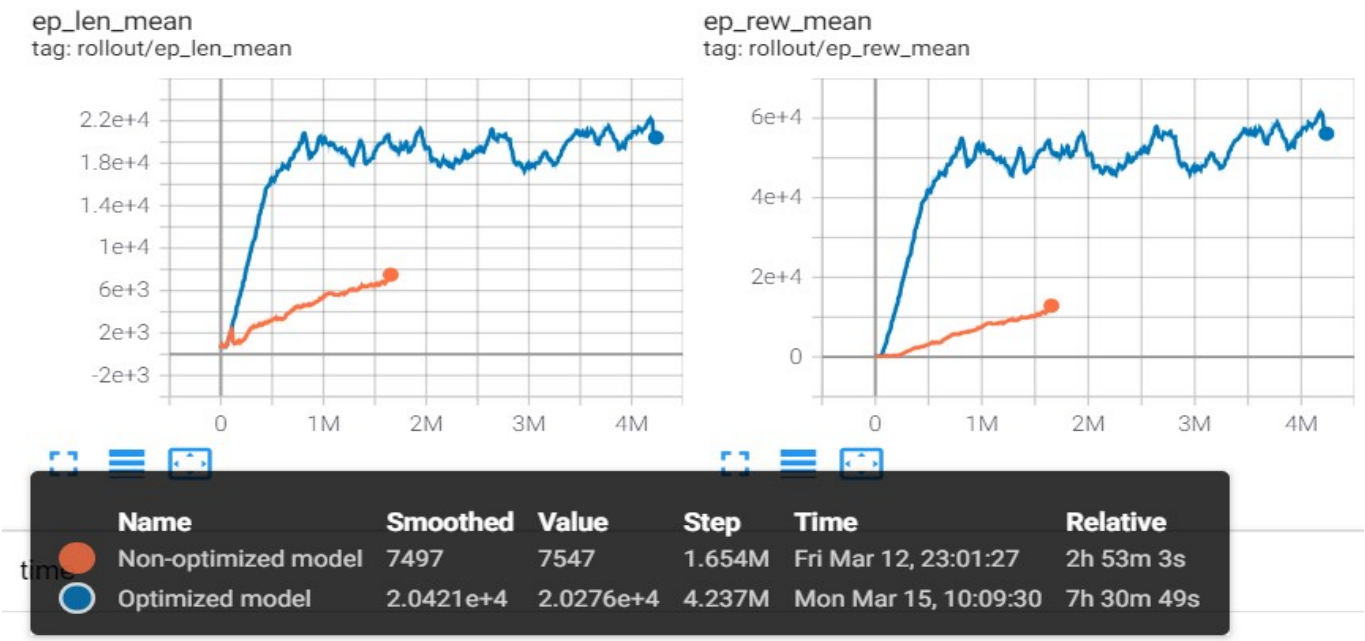


Σχήμα 14: Σύγκριση επίδοσης αλγορίθμου QR-DQN αλγορίθμου με και χωρίς στοχαστικότητα

Παρατηρούμε ότι μετά από περίπου 3 ώρες εκπαίδευσης η εισαγωγή στοχαστικότητας δεν βελτιώνει περαιτέρω την επίδοση του μοντέλου μας. Για τον λόγο αυτό εκπαιδεύουμε στο επόμενο βήμα το βέλτιστο μοντέλο σε περιβάλλον χωρίς sticky actions (v4).

Εκπαίδευση τελικού βελτιστοποιημένου μοντέλου

Συνεχίζουμε το training του μοντέλου μας για άλλες 18 ώρες (21 συνολικά). Λόγω του περιορισμένου χρόνου που μπορεί να παραμείνει ενεργό ένα Kaggle Session, η εκπαίδευση αυτή πραγματοποιείται τμηματικά αξιοποιώντας τις εντολές save και load, όπως εξηγείται στο notebook της εκφώνησης. Στο επόμενο διάγραμμα παρουσιάζεται τόσο η αρχική τρίωρη εκπαίδευση του όσο και αυτή των τελευταίων 7,5 ωρών (από τις 21 συνολικά).



Σχήμα 15: Αρχικό και βελτιστοποιημένο μοντέλο

Παρατηρούμε πως πετυχαίνουμε μια πολύ σημαντική αύξηση της απόδοσης του αρχικού μοντέλου μας, καθώς το ep_rew_mean αυξάνεται από 13.000 σε περίπου 60.000. Εξετάζουμε την επίδοση του μοντέλου στο περιβάλλον ελέγχου και σημειώνουμε σκορ ίσο με **102.675**! Η μεγάλη αυτή απόκλιση μεταξύ training και test reward οφείλεται στο γεγονός ότι το μοντέλο έχει μεγάλη διακύμανση, καθώς επιδέχεται περαιτέρω εκπαίδευση για να σταθεροποιηθεί γύρω από ένα υψηλότερο reward. Ωστόσο, λόγω των περιορισμένων πόρων GPU που διατίθενται μέσω του kaggle δεν είναι εφικτή η συνέχιση της εκπαίδευσης του. Παρόλα αυτά, το score που σημειώνουμε είναι εξαιρετικά υψηλό και προσεγγίζει ορισμένα state of the art μοντέλα. Δοκιμάσαμε να παίξουμε το συγκεκριμένο παιχνίδι και το σκορ που πετύχαμε ήταν πολύ χαμηλότερο (κάτω από 5.000) από αυτό του μοντέλου, ωστόσο δεν διαθέταμε κάποια προηγούμενη εμπειρία. Αξίζει να αναφερθεί ότι, με βάση τη σελίδα αυτή, το υψηλότερο σκορ που έχει σημειωθεί από άνθρωπο υπερβαίνει το 1,5 εκατομμύριο, επίδοση πολύ μεγαλύτερη από οποιοδήποτε δίκτυο. Παρόλα αυτά, ένα σκορ πάνω από 100.000 είναι εξαιρετικά δύσκολο να επιτευχθεί από κάποιον μη έμπειρο παίχτη.