
Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα



Εργασία 4. Παιζόντας Atari με Βαθιά Ενισχυτική Μάθηση.

Σκουρτσίδης Γιώργος
03114307

Κωνσταντίνος Γεωργάς
03116755

Γιάννης Βονδικάκης
03113186

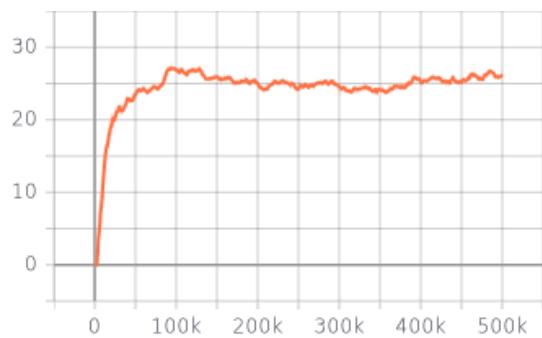


Ξεκινήστε με έναν πράκτορα χωρίς καθόλου μάθηση (random agent). Ποια είναι η μέση ανταμοιβή του (score) σε ένα περιβάλλον test;

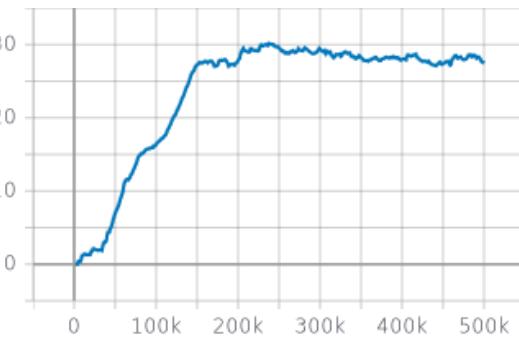
Ορίσαμε έναν random agent με τη μέθοδο που μας προτάθηκε από τους διδάσκοντες. Στη συνέχεια κάναμε evaluate τον agent. Η μέση ανταμοιβή του ήταν μηδενική.

Χρησιμοποιώντας το TensorBoard δείτε ποιοι αλγόριθμοι RL είναι πιθανότερο να έχουν καλή επίδοση.

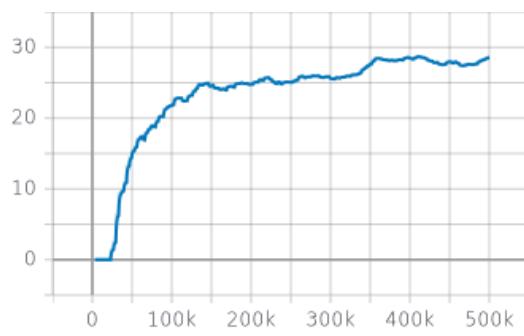
A2C:



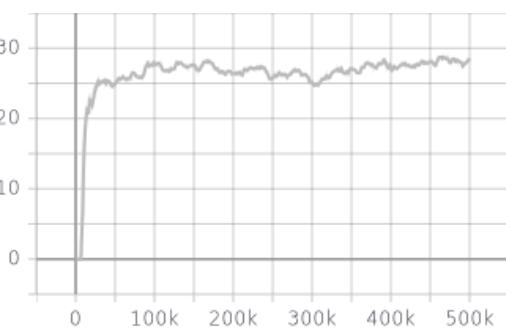
Εικόνα 1: v4



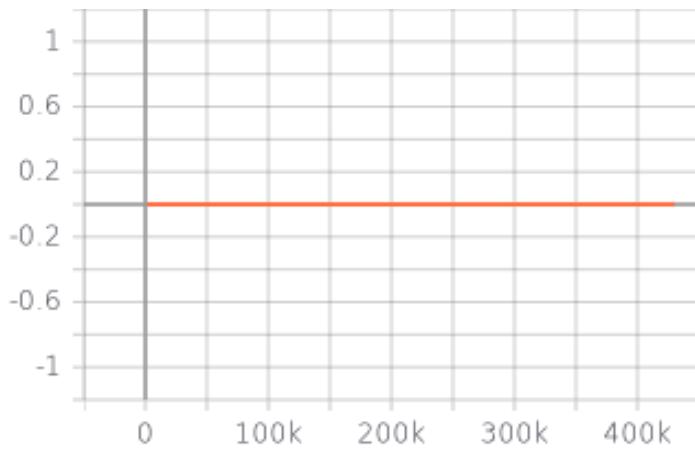
Εικόνα 2: v4 deterministic



Εικόνα 3: v0-NoFrameskip

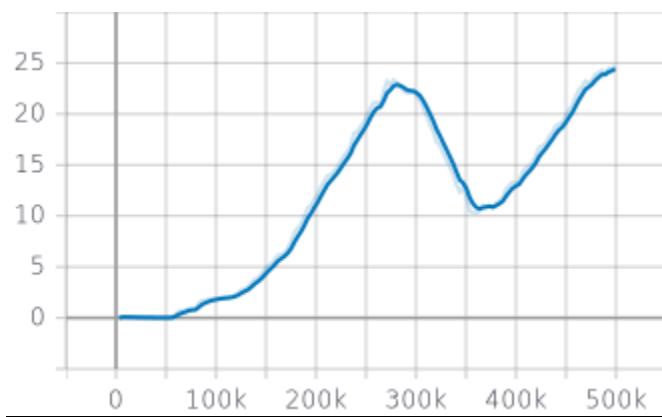


Εικόνα 4: v0 deterministic

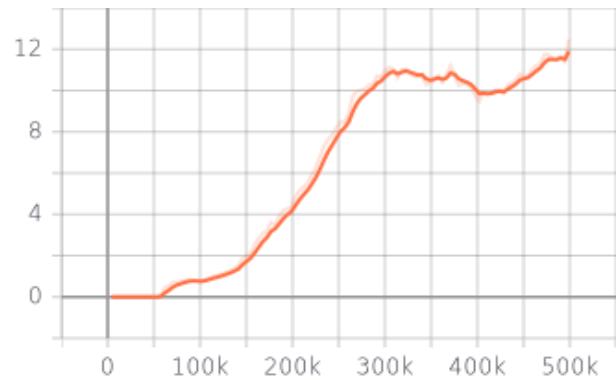


Εικόνα 5: v0

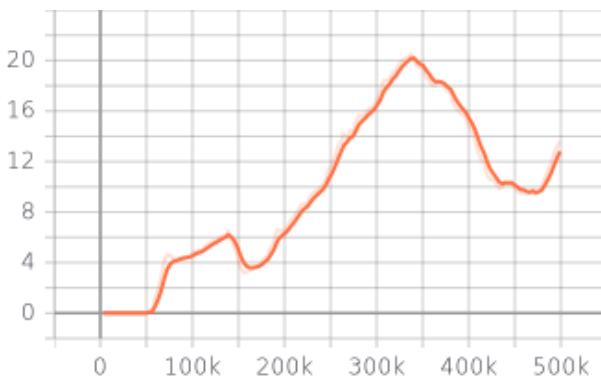
DQN



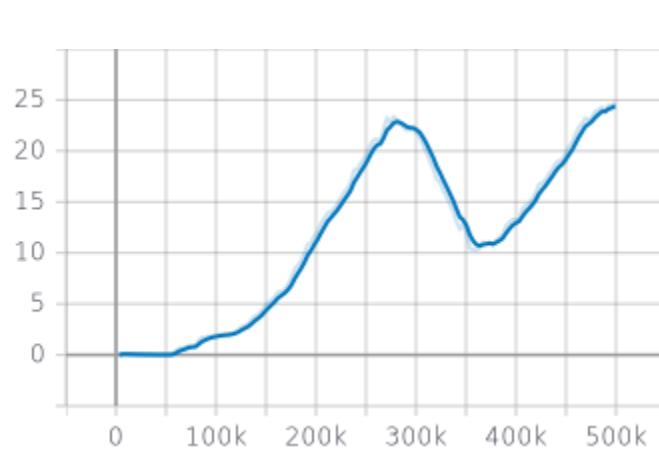
Εικόνα 6: v0



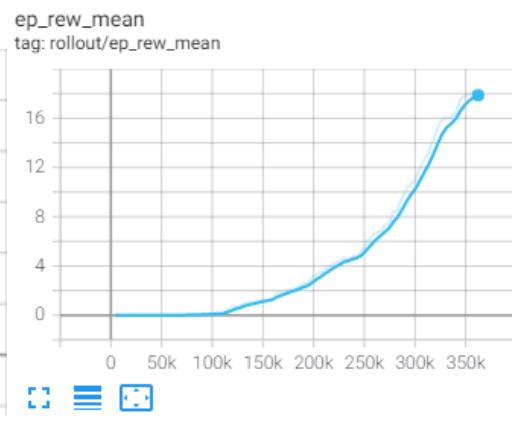
Εικόνα 7: v4 No Frameskip



Εικόνα 8: v4 Deterministic

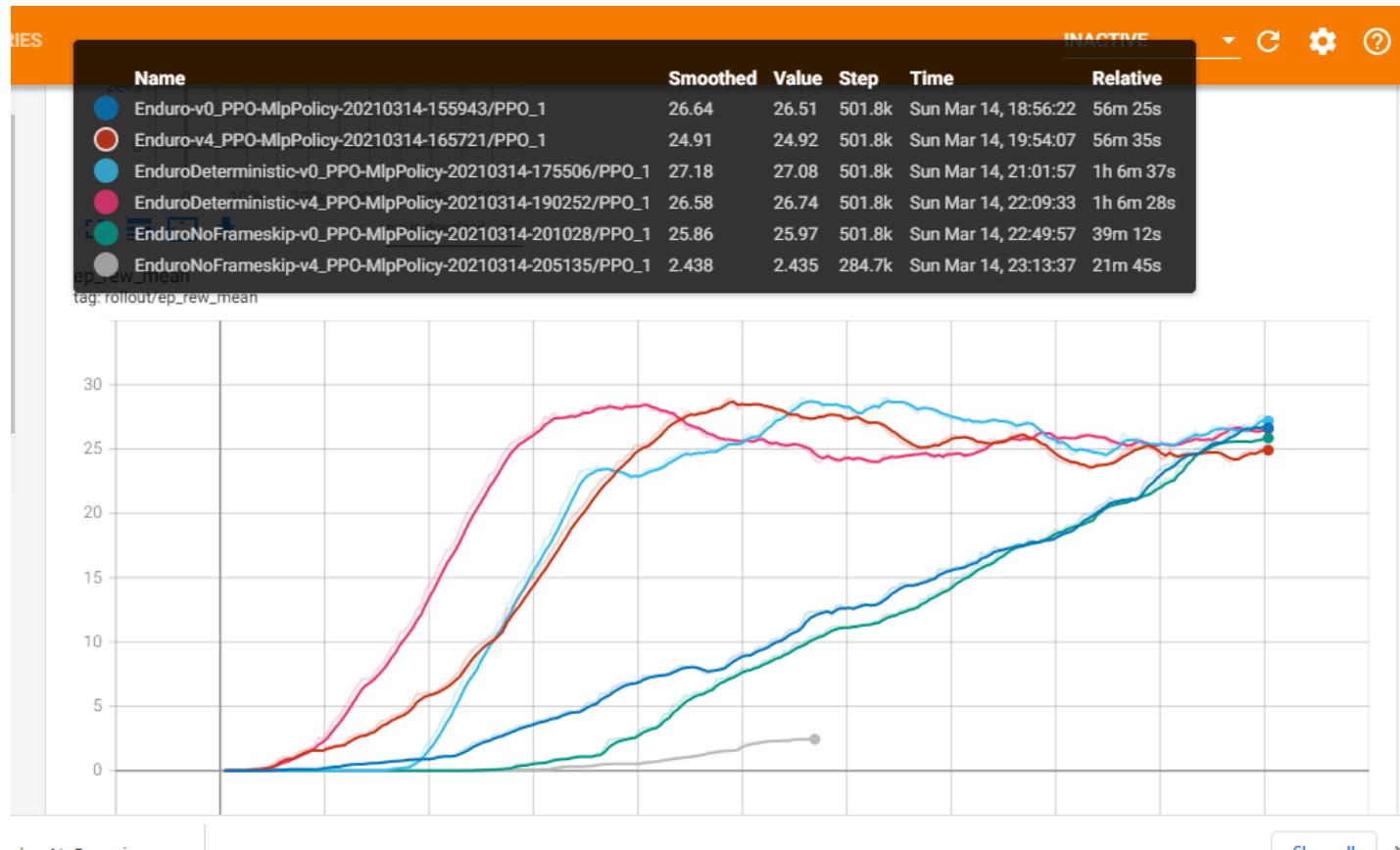


Εικόνα 9: v4-No Frameskip



Εικόνα 10: v0-Deterministic

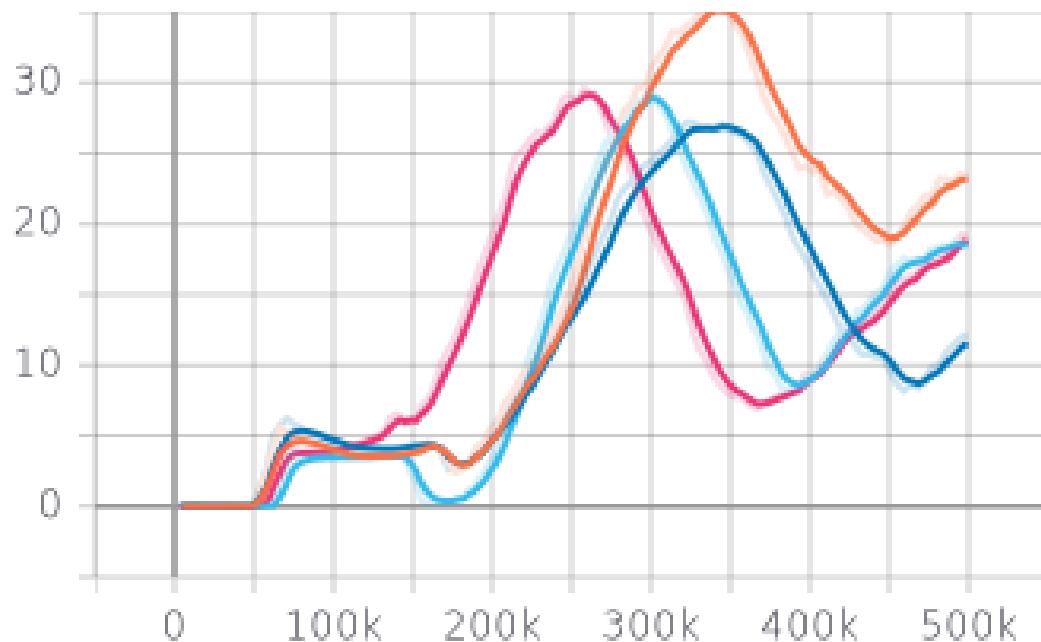
PPO



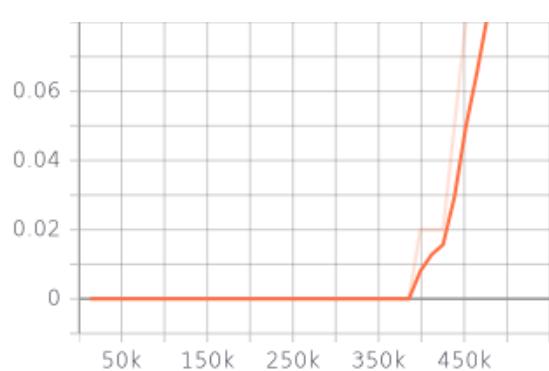
Εικόνα 11: Συγκεντρωτικό γράφημα για το PPO

Το “EnduroNoFrameskip-v4” το σταματήσαμε καθώς βλέπαμε πως η απόδοσή του ήταν κατώτερη από τα άλλα.

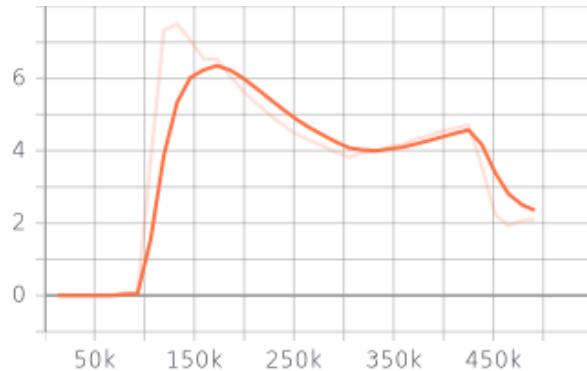
DRDQN



Εικόνα 12: Μπλε: v4, Γαλάζιο : v0 Deterministic, Ροζ : v4 Deterministic, Πορτοκαλί : v0



Εικόνα 13: v4 No Frameskip



Εικόνα 14:: v0 No Frameskip

Αποτελέσματα στο test environment

PPO
Enduro-v0, Eval reward: 28.6 (+/-8.99110671719561)
Enduro-v4, Eval reward: 31.2 (+/-15.612815249018995)
EnduroDeterministic-v0, Eval reward: 26.5 (+/-9.135097153287424)
EnduroDeterministic-v4, Eval reward: 24.3 (+/-8.414867794564572)
EnduroNoFrameskip-v0, Eval reward: 26.6 (+/-14.037093716293269)
EnduroNoFrameskip-v4, Eval reward: 37.7 (+/-11.730728877610291)
A2C
Enduro-v0, Eval reward: 0.0 (+/-0.0)
Enduro-v4, Eval reward: 31.2 (+/-15.612815249018995)
EnduroDeterministic-v0, Eval reward: 26.5 (+/-9.135097153287424)
EnduroDeterministic-v4, Eval reward: 24.3 (+/-8.414867794564572)
EnduroNoFrameskip-v0, Eval reward: 28.1 (+/-14.397569239284804)
EnduroNoFrameskip-v4, Eval reward: 0.0 (+/-0.0)
QRDQN
Enduro-v0, Eval reward: 42.3 (+/-20.885641000457706)
Enduro-v4, Eval reward: 25.8 (+/-12.21310771261762)
EnduroDeterministic-v0, Eval reward: 19.4 (+/-10.650821564555478)
EnduroDeterministic-v4, Eval reward: 24.3 (+/-10.574024777727733)
EnduroNoFrameskip-v0, Eval reward: 1.1 (+/-1.4456832294800963)
EnduroNoFrameskip-v4, Eval reward: 0.5 (+/-0.5)

Εικόνα 15: Συγκεντρωτικά αποτελέσματα

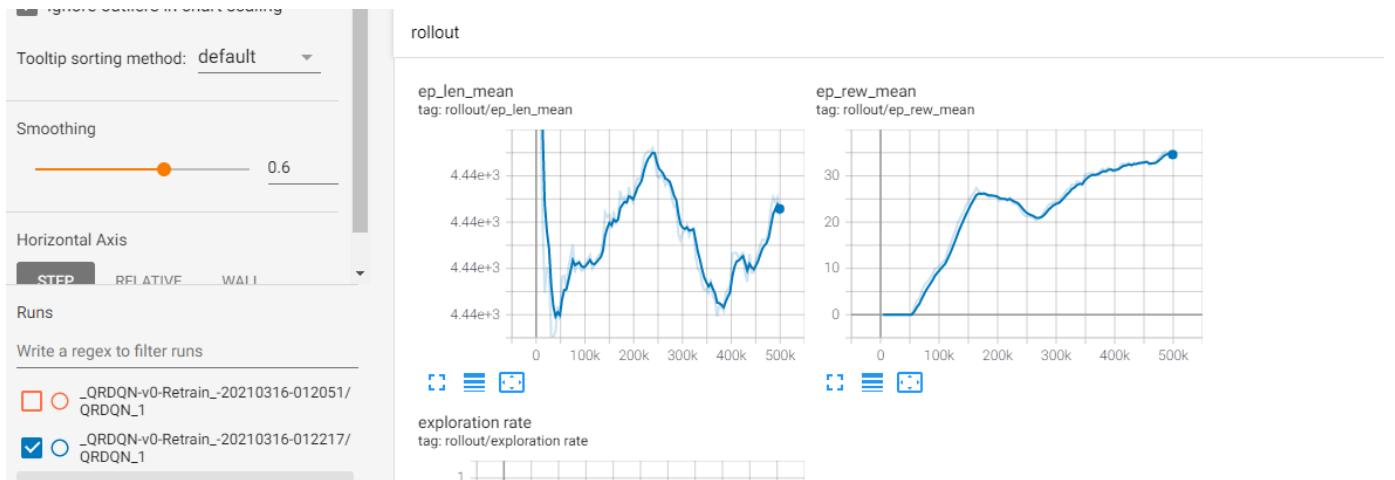
Δείτε αν η σχετική απόδοση των αλγορίθμων διαφοροποιείται μεταξύ Deterministic, NoFrameskip και "σκέτο".

Συγκρίνοντας τα διαγράμματα βλέπουμε πως δεν μπορούμε με σιγουριά να αποφανθούμε για την ανωτερότητα κάποιου εκ των Deterministic, NoFrameskip και "σκέτο". Σε κάθε μοντέλο (και λαμβάνοντας υπόψη την στοχαστικότητα των πειραμάτων) λαμβάνουμε αρκετά διαφορετικά αποτελέσματα. Θα προχωρήσουμε στο επόμενο βήμα κρατώντας μόνο τους συνδυασμούς που μας έδωσαν τα καλύτερα αποτελέσματα. Γενικά εκ του αποτελέσματος παρατηρήσαμε ελάχιστα καλύτερη αποδοτικότητα στα στοχαστικά μοντέλα (-v0), με κόστος όμως τον υψηλότερο χρόνο εκπαίδευσης.

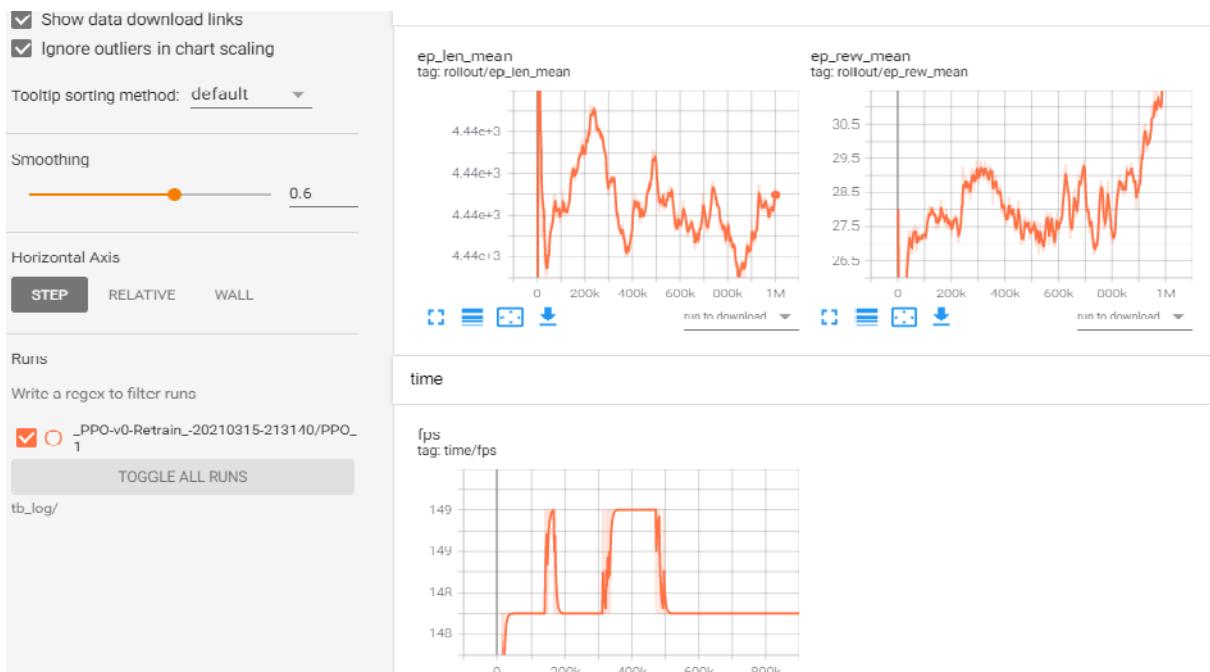
Βελτιώστε όσο είναι δυνατόν τους αλγόριθμους που ξεχωρίζουν.

Βελτιστοποιήσαμε περαιτέρω τους αλγόριθμους «PPO-v0», «QRDQN -v0» και «PPO-NoFrameSkip-v4», χρησιμοποιώντας την τεχνική της συνέχισης της εκπαίδευσης. Δοκιμάσαμε να βελτιστοποιήσουμε τις παραμέτρους «gamma», «tau» και «learning rate». Επίσης δοκιμάσαμε να τρέξουμε τους αλγορίθμους και με CnnPolicy, με μη ικανοποιητικά αποτελέσματα.

Τα αποτελέσματα παρουσιάζονται παρακάτω.

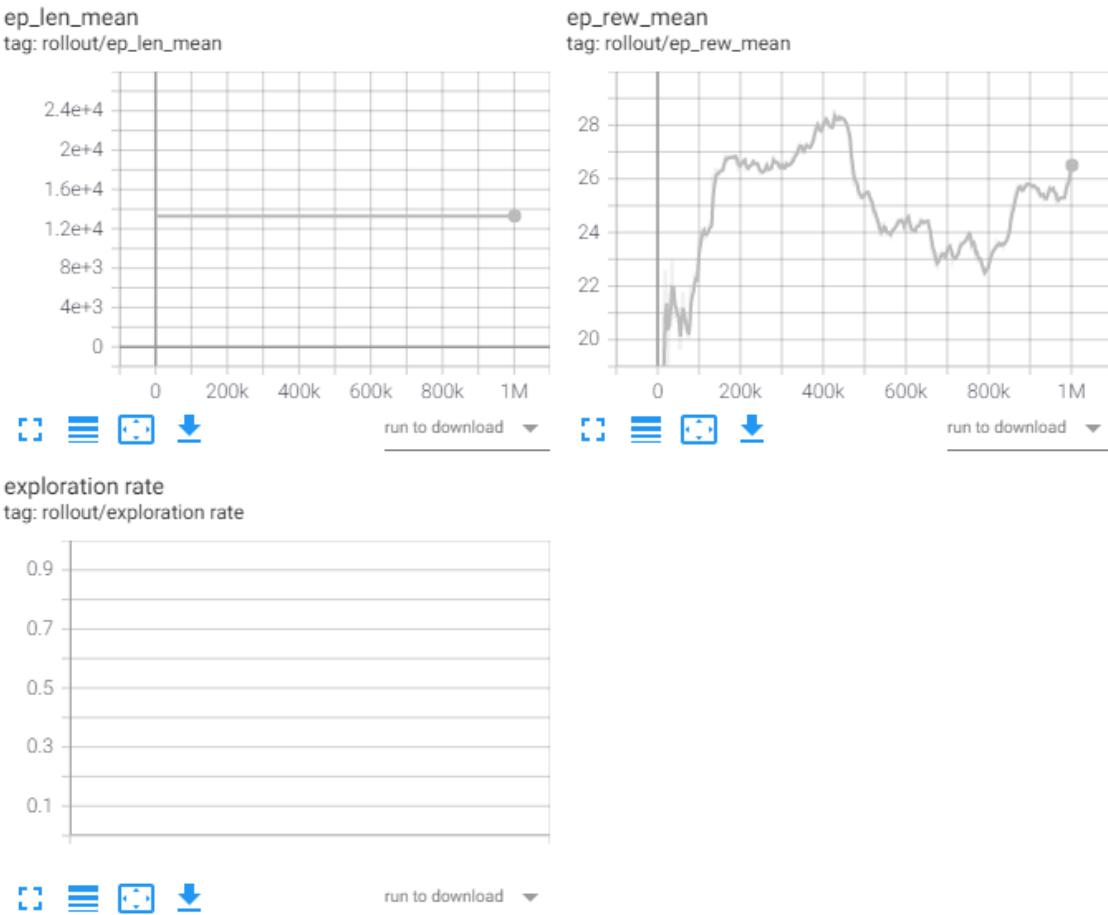


Εικόνα 16: QRDQN →v0



Εικόνα 17: PPO-v0

To video από το μοντέλο PPO-v0 υπάρχει στον παρακάτω σύνδεσμο: [PPO v0](#)



Εικόνα 18: PPO NoFrameSkip-v4

Από τα παραπάνω μοντέλα καλύτερο αναδεικνύεται το QRDQN -v0. Τα αποτελέσματα στο test environment παρουσιάζονται παρακάτω.

```
▶ test_env = make_atari_env(atari_env_name, n_envs=1, seed=0)
# Frame-stacking with 4 frames
test_env = VecFrameStack(test_env, n_stack=4)

mean_reward, std_reward = evaluate_policy(model, test_env, n_episodes=10)
print(f"Eval reward: {mean_reward} (+/-{std_reward})")
model.save("QRDQN-v0-5m")
```

Eval reward: 42.7 (+/-24.686230980042296)

+ Code

+ Markdown

Εικόνα 19: Evaluate model in test Env

Από τα state-of-the-art μοντέλα τα αποτελέσματα μας βρίσκονται αρκετά χαμηλότερα, σε νούμερα που δυστυχώς δεν είναι συγκρίσιμα.

Ο καλύτερός μας agent είναι ο «QRDQN –v0». Ο agent μπορεί να φορτωθεί στο notebook που αποστέλλουμε, καθώς και να γίνει evaluate ή να εκπαιδευτεί/βελτιστοποιηθεί περαιτέρω. Το βίντεο της δράσης του είναι το ακόλουθο: [DRDQN-v0](#)

Παρατηρούμε πως ο agent μόλις έχει αρχίσει να μαθαίνει να αποφεύγει εμπόδια, πράγμα που αδυνατούσαν να κάνουν τα περισσότερα άλλα μοντέλα . Εάν είχαμε περισσότερο χρόνο ή μεγαλύτερη υπολογιστική ισχύ για διερεύνηση βέλτιστων παραμέτρων πιθανώς να καταλήγαμε σε αρκετά βελτιωμένα αποτελέσματα, είτε στο [DRDQN-v0](#), είτε στο [PPO v0](#) μοντέλο.

Προκειμένου να έχουμε ένα μέτρο σύγκρισης του αλγορίθμου σε σχέση με τον ανθρώπινο παράγοντα, δοκιμάσαμε να παίξουμε και εμείς το παιχνίδι Enduro. Αρχικά , καθώς δεν γνωρίζαμε καθόλου πως να παίξουμε, τα αποτελέσματά μας ήταν συγκρίσιμα με τους μη-βελτιστοποιημένους αλγορίθμους. Ύστερα από λίγα λεπτά «εκπαίδευσης» καταφέραμε σκορ 1.000 . Οι καλύτεροί μας αλγόριθμοι πετυχαίνουν σκορ 700-770, δηλαδή είναι συγκρίσιμα αποτελέσματα με έναν αρχάριο παίχτη του παιχνιδιού.