



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Σύγκριση αλγορίθμων μηχανικής μάθησης για την εκπαίδευση ευφυών πρακτόρων σε περιβάλλον παιχνιδιού

Comparison of machine learning algorithms for the training of intelligent agents in a game environment

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Τσιάλιος  
Αριθμός Μητρώου: 1072868

Επιβλέπων  
Κυριάκος Σγάρμπας, Αναπληρωτής Καθηγητής

Πάτρα  
Σεπτέμβριος 2024



Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

©2024 —Με την επιφύλαξη παντός δικαιώματος

Το σύνολο της εργασίας αποτελεί πρωτότυπο έργο, παραχθέν από τον Γεώργιο Τσιάλιο, και δεν παραβιάζει δικαιώματα τρίτων καθ' οιονδήποτε τρόπο. Αν η εργασία περιέχει υλικό, το οποίο δεν έχει παραχθεί από τον ίδιο, αυτό είναι ευδιάκριτο και αναφέρεται ρητώς εντός του κειμένου της εργασίας ως προϊόν εργασίας τρίτου, σημειώνοντας με παρομοίως σαφή τρόπο τα στοιχεία ταυτοποίησής του, ενώ παράλληλα βεβαιώνει πως στην περίπτωση χρήσης αυτούσιων γραφικών αναπαραστάσεων, εικόνων, γραφημάτων κ.λπ., έχει λάβει τη χωρίς περιορισμούς άδεια του κατόχου των πνευματικών δικαιωμάτων για την συμπερίληψη και επακόλουθη δημοσίευση του υλικού αυτού.



Γεώργιος Τσιάλιος



## ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα  
**Σύγκριση αλγορίθμων μηχανικής μάθησης για την εκπαίδευση  
ευφυών πρακτόρων σε περιβάλλον παιχνιδιού**  
του φοιτητή του τμήματος Ηλεκτρολόγων Μηχανικών & Τεχνολογίας  
Υπολογιστών

### Γεωργίου Τσιάλιου του Ιωάννη

Αριθμός Μητρώου: 1072868

παρουσιάστηκε δημόσια στο τμήμα Ηλεκτρολόγων Μηχανικών &  
Τεχνολογίας Υπολογιστών στις

13/9/2024

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Κυριάκος Σγάρμπας, Αναπληρωτής Καθηγητής, ΤΗΜ&ΤΥ (επιβλέπων)  
----- (μέλος επιτροπής)  
----- (μέλος επιτροπής)

Ο Επιβλέπων

Ο Διευθυντής του Τομέα

Κυριάκος Σγάρμπας  
Αναπληρωτής Καθηγητής

Μιχαήλ Λογοθέτης  
Αναπληρωτής Καθηγητής



## Σύνοψη

Στη σημερινή εποχή, το επιστημονικό πεδίο της Τεχνητής Νοημοσύνης αποτελεί ένα από τα πιο ραγδαία αναπτυσσόμενα ερευνητικά αντικείμενα παγκοσμίως. Με πιο πρόσφατο παράδειγμα την ανάπτυξη των μεγάλων γλωσσικών μοντέλων (LLMs) όπως το ChatGPT της OpenAI, η Τεχνητή Νοημοσύνη παρεισφρύει ολοένα και περισσότερο στη ζωή των ανθρώπων, παρέχοντας εφαρμογές που λύνουν προβλήματα της καθημερινότητας με υπεράνθρωπη ακρίβεια και ταχύτητα. Στον πυρήνα των εφαρμογών αυτών βρίσκονται συχνά αλγόριθμοι Μηχανικής Μάθησης, ενός υποπεδίου της Τεχνητής Νοημοσύνης. Οι συγκεκριμένοι αλγόριθμοι εξετάζονται συνήθως, πρώτα σε δοκιμαστικά περιβάλλοντα, όπως παιχνίδια, όπου η προσομοίωση της πραγματικότητας είναι εύκολη και ακίνδυνη.

Στο πλαίσιο αυτό, στόχος της παρούσας διπλωματικής εργασίας είναι η εκπαίδευση πρακτόρων τεχνητής νοημοσύνης σε ένα απλό παιχνίδι, χρησιμοποιώντας διαφορετικούς αλγορίθμους και η σύγκριση τους σε όρους χρόνου εκπαίδευσης και τελικής επίδοσης. Συγκεκριμένα, αναπτύχθηκε ένα παιχνίδι στο οποίο ο πράκτορας καλείται να παρκάρει ένα αυτοκίνητο σε μία τυχαία θέση στάθμευσης. Οι αλγόριθμοι που εξετάστηκαν ανήκουν στην υποκατηγορία της Μηχανικής Μάθησης που ονομάζεται Ενισχυτική Μάθηση και είναι οι εξής: Q-learning, Proximal Policy Optimization (PPO), Soft Actor Critic (SAC), Deep Deterministic Policy Gradient (DDPG) και Twin Delayed Deep Deterministic Policy Gradient (TD3).

Ο κώδικας που αναπτύχθηκε για την υλοποίηση του παιχνιδιού και την εκπαίδευση των πρακτόρων είναι ελεύθερα διαθέσιμος στον παρακάτω σύνδεσμο: [GitHub Repository](#).

**Λέξεις-κλειδιά:** Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Βαθιά Ενισχυτική Μάθηση, Νευρωνικά Δίκτυα, Σύγκριση αλγορίθμων, Αυτοδηγούμενα οχήματα, OpenAI Gymnasium, Stable-Baselines 3



## Abstract

In the modern era, the scientific field of Artificial Intelligence is one of the most rapidly evolving research areas worldwide. Artificial Intelligence is increasingly entering people's lives, by offering applications that solve everyday problems with superhuman accuracy and speed. A recent example of this, is the development of Large Language Models (LLMs) like OpenAI's ChatGPT. At the core of these applications are often Machine Learning algorithms, a subfield of Artificial Intelligence. These algorithms are usually tested first in experimental environments, such as games, where simulating reality is easy and safe.

In this context, the goal of this thesis is to train Artificial Intelligence agents in a simple game using different algorithms and compare them in terms of training time and final performance. More specifically, a game was developed in which the agent is tasked with parking a car in a random parking spot. The following algorithms -which belong to the Machine Learning subcategory called Reinforcement Learning- were used: Q-learning, Proximal Policy Optimization (PPO), Soft Actor Critic (SAC), Deep Deterministic Policy Gradient (DDPG) and Twin Delayed Deep Deterministic Policy Gradient (TD3).

The code developed for the implementation of the game and the training of the agents is publicly available at the following link: [GitHub Repository](#).

**Keywords:** Artificial Intelligence, Machine Learning, Deep Reinforcement Learning, Neural Networks, Algorithm comparison, Self-driving cars, OpenAI Gymnasium, Stable-Baselines 3



## Ευχαριστίες

Ολοκληρώνοντας την ακαδημαϊκή μου σταδιοδρομία, θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν, έμπρακτα αλλά και όχι μόνο, τα τελευταία πέντε έτη των σπουδών μου.

Αρχικά, επιθυμώ να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας, κο Κυριάκο Σγάρμπα, ο οποίος μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον ερευνητικό πεδίο. Ακόμα, η εμπιστοσύνη που μου έδειξε και η καθοδήγηση του έπαιξαν καθοριστικό ρόλο στην ομαλή εκπόνηση της παρούσας εργασίας.

Στη συνέχεια, θα ήθελα να ευχαριστήσω θερμά την οικογένεια και τους φίλους μου για την συνεχή στήριξη που μου προσφέρουν. Ειδική αναφορά επιθυμώ να κάνω σε δύο άτομα, στα οποία οφείλω, σε μεγάλο βάθμο, τη μέχρι τώρα πορεία μου. Αρχικά, ευχαριστώ τον πατέρα μου, Ιωάννη, για τη μόνψη υποστήριξη και την ανεκτίμητη συμπαράσταση του. Έπειτα, ευχαριστώ τον εξαιρετικό συνάδελφο και φίλο, Χρήστο Κατσανδρή, ο οποίος με ενέπνευσε σημαντικά, να γίνω καλύτερος. Η αδιάλειπτη προθυμία του να βοηθήσει αποδείχθηκε πολύτιμη πολλές φορές, ενώ η συνεργασία μας σε διάφορα μαθήματα και εργασίες ήταν χαρά και τιμή μουν.



# Περιεχόμενα

<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Στόχος . . . . .	1
1.2 Διάρθρωση Εργασίας . . . . .	2
<b>2 Επισκόπηση του χώρου</b>	<b>3</b>
2.1 Αυτόνομα οχήματα . . . . .	3
2.1.1 Ορισμός . . . . .	3
2.1.2 Αρχές λειτουργίας . . . . .	3
2.1.3 Πλεονεκτήματα και Μειονεκτήματα . . . . .	5
2.1.4 Η κατάσταση σήμερα . . . . .	6
2.2 Προηγούμενη έρευνα . . . . .	7
<b>3 Βασική Θεωρία</b>	<b>9</b>
3.1 Τεχνητή Νοημοσύνη . . . . .	9
3.1.1 Ορισμός . . . . .	9
3.1.2 Μοντέλα και Πράκτορες τεχνητής νοημοσύνης . . . . .	9
3.1.3 Ιστορική Εξέλιξη . . . . .	10
3.1.4 Κατηγορίες . . . . .	11
3.1.5 Εφαρμογές . . . . .	14
3.2 Μηχανική Μάθηση . . . . .	15
3.2.1 Ορισμός . . . . .	15
3.2.2 Κατάταξη πεδίου . . . . .	16
3.2.3 Κατηγορίες . . . . .	17
3.3 Ενισχυτική Μάθηση . . . . .	19
3.3.1 Γενική επισκόπηση . . . . .	19
3.3.2 Βασικές Έννοιες και Ορολογία . . . . .	23
3.3.3 Κατηγορίες αλγορίθμων . . . . .	33
3.3.4 Ο αλγόριθμος $Q$ -learning . . . . .	39
3.4 Βαθιά Ενισχυτική Μάθηση . . . . .	47
3.4.1 Ορισμός και Χαρακτηριστικά . . . . .	47
3.4.2 Τεχνητά Νευρωνικά Δίκτυα . . . . .	48

3.4.3	Ο αλγόριθμος PPO . . . . .	60
3.4.4	Ο αλγόριθμος DDPG . . . . .	63
3.4.5	Ο αλγόριθμος TD3 . . . . .	65
3.4.6	Ο αλγόριθμος SAC . . . . .	68
3.5	Σύνοψη . . . . .	70
<b>4</b>	<b>Το παιχνίδι</b>	<b>71</b>
4.1	Αιτιολογία κατασκεύης παιχνιδιού . . . . .	71
4.2	Κανόνες παιχνιδιού . . . . .	75
4.2.1	Βασικοί κανόνες . . . . .	75
4.2.2	Γενίκευση . . . . .	77
4.2.3	Επίπεδα δυσκολίας . . . . .	78
4.3	Κατασκευή παιχνιδιού . . . . .	79
4.3.1	Εργαλεία . . . . .	79
4.3.2	Προσωμοίωση φυσικής . . . . .	80
4.3.3	Ενδιαφέρουσες υλοποιήσεις . . . . .	82
4.3.4	Αδυναμίες . . . . .	89
<b>5</b>	<b>Εκπαίδευση</b>	<b>93</b>
5.1	Στόχοι . . . . .	94
5.2	Εργαλεία . . . . .	94
5.2.1	Gymnasium . . . . .	95
5.2.2	Numpy . . . . .	96
5.2.3	Stable-Baselines3 . . . . .	96
5.2.4	TensorBoard . . . . .	97
5.2.5	Colaboratory . . . . .	98
5.2.6	Github . . . . .	99
5.3	Υπολογιστικοί πόροι . . . . .	99
5.4	Στατιστικά . . . . .	100
5.5	Δυσκολίες Ενισχυτικής Μάθησης . . . . .	102
5.5.1	Χρόνος εκπαίδευσης . . . . .	102
5.5.2	Αποσφαλμάτωση . . . . .	102
5.5.3	Υπερ-παραμέτροι . . . . .	104
5.5.4	Σχεδίαση συνάρτησης ανταμοιβής . . . . .	104
5.5.5	Αστάθεια της εκπαίδευσης . . . . .	108
5.6	Καλές πρακτικές . . . . .	109
5.6.1	Επιλογή αλγορίθμου . . . . .	109
5.6.2	Επιλογή πολιτικής αλγορίθμου . . . . .	110

5.6.3	Μοντελοποίηση προβλήματος	110
5.6.4	Παράμετροι Εκπαίδευσης	113
5.6.5	Κανονικοποιήσεις τιμών	113
5.6.6	Παράκαμψη βημάτων	114
5.6.7	Επίπεδα δυσκολίας και Κλιμακωτή Μάθηση	115
5.6.8	Παρακολούθηση μετρικών	116
5.6.9	Διατήρηση αρχείου εκπαίδεύσεων	116
5.7	Μοντελοποίηση προβλήματος	117
5.7.1	Αρχιτεκτονική νευρωνικών δικτύων	117
5.7.2	Συνάρτηση ανταμοιβής	121
5.8	Εκπαίδευσης με τον αλγόριθμο Q-Learning	126
5.8.1	Επισκόπηση εκπαίδεύσεων	126
5.8.2	Καλύτερες εκπαίδευσης	128
5.9	Εκπαίδευσης με τον αλγόριθμο PPO	132
5.9.1	Επισκόπηση εκπαίδεύσεων	132
5.9.2	Καλύτερες εκπαίδευσης	133
5.10	Εκπαίδευσης με τον αλγόριθμο SAC	135
5.10.1	Επισκόπηση εκπαίδεύσεων	135
5.10.2	Καλύτερες εκπαίδευσης	135
5.11	Εκπαίδευσης με τον αλγόριθμο TD3	136
5.11.1	Επισκόπηση εκπαίδεύσεων	136
5.11.2	Καλύτερες εκπαίδευσης	137
5.12	Εκπαίδευσης με τον αλγόριθμο DDPG	138
5.12.1	Επισκόπηση εκπαίδεύσεων	138
5.12.2	Καλύτερες εκπαίδευσης	139
<b>6</b>	<b>Αξιολόγηση αποτελεσμάτων</b>	<b>141</b>
6.1	Σύγκριση αλγορίθμων ως προς τον χρόνο εκπαίδευσης	141
6.1.1	Επίπεδο δυσκολίας 3 - Άμεση στάθμευση	142
6.1.2	Επίπεδο δυσκολίας 4 - Κανονική στάθμευση	143
6.2	Σύγκριση αλγορίθμων ως προς την απόδοση	143
6.2.1	Επίπεδο δυσκολίας 3 - Άμεση στάθμευση	144
6.2.2	Επίπεδο δυσκολίας 4 - Κανονική στάθμευση	147
6.3	Συμπεράσματα	150
<b>7</b>	<b>Μελλοντικές Βελτιώσεις</b>	<b>153</b>
<b>8</b>	<b>Βιβλιογραφία</b>	<b>155</b>



## **Κατάλογος πινάκων**

5.1	Πίνακας τεχνικών χαρακτηριστικών διαθέσιμων υπολογιστικών συστημάτων . . . . .	99
5.2	Πίνακας καταλληλότητας αλγορίθμων ενισχυτικής μάθησης . . . . .	110



# Κατάλογος σχημάτων

2.1	Επίπεδα αυτοματισμού αυτόνομων οχημάτων (Society of Automobile Engineers 2021).	4
2.2	Τεχνολογίες αυτόνομων οχημάτων (University of Michigan Center for Sustainable Systems 2023) . . . . .	5
3.1	Κατηγορίες τεχνητής νοημοσύνης (Lateef 2024). . . . .	12
3.2	Ιεραρχία πεδίων τεχνητής νοημοσύνης (Zand κ.ά. 2022). . . . .	16
3.3	Κατηγορίες μηχανικής μάθησης (Stewart 2023). . . . .	17
3.4	Κύκλος Ενισχυτικής Μάθησης (Lee 2019). . . . .	24
3.5	Παράδειγμα Διαδικασίας Απόφασης Μαρκόβ (Alvarez 2017)	25
3.6	Το δίλημμα Εξερεύνησης - Εκμετάλλευσης (Parkinson 2019). . . . .	31
3.7	Ταξινόμηση αλγορίθμων ενισχυτικής μάθησης (OpenAI 2018). . . . .	34
3.8	Εκτενέστερη ταξινόμηση αλγορίθμων ενισχυτικής μάθησης (Prijono 2020). . . . .	38
3.9	Πίνακας $Q$ για την αποθήκευση των τιμών της συνάρτησης $Q(s, a)$ (Baeldung 2023). .	40
3.10	Η τεχνική $\varepsilon$ -greedy (Baeldung 2023). . . . .	42
3.11	Το παιχνίδι Frozen Lake της βιβλιοθήκης OpenAI Gymnasium.	44
3.12	Αποτελέσματα εκπαίδευσης στο παιχνίδι Frozen Lake (Szymanski 2018). . . . .	45
3.13	Κλασική Ενισχυτική Μάθηση εναντίον Βαθιάς Ενισχυτικής Μάθησης (Quang 2024). .	48
3.14	Σύγκριση βιολογικού και τεχνητού νευρώνα (Karpathy 2016). . . . .	49
3.15	Γραφική παράσταση της συνάρτησης Tanh. . . . .	51
3.16	Γραφική παράσταση της συνάρτησης ReLU. . . . .	52
3.17	Παράδειγμα τεχνητού νευρωνικού δικτύου (Comsol 2023). . . . .	53
3.18	Αλληλεπίδραση δικτύων Δράστη-Κριτή (Sutton και Barto 2018). . . . .	60
3.19	Συχνότητα χρήσης αλγορίθμων ενισχυτικής μάθησης σε επιστημονικές δημοσιεύσεις (Papers With Code 2024). . . . .	61
3.20	Τύποι προβλημάτων αλγορίθμου PPO (Papers With Code 2024) . . . . .	61
3.21	Ψευδοκώδικας του αλγορίθμου DDPG. . . . .	65
4.1	Επικοινωνία πράκτορα με το παιχνίδι Trackmania (Yosh 2022). . . . .	72
4.2	Στιγμότυπο του παιχνιδιού «Parking Game». . . . .	75
4.3	Εργαλεία κατασκευής παιχνιδιού. . . . .	80
4.4	Παράδειγμα εικόνας για τη δημιουργία μάσκας του χάρτη. . . . .	83

4.5 Ορθογώνιο αυτοκινήτου έναντι εικόνας αυτοκινήτου. Παρατηρούμε πως η ανίχνευση σύγκρουσης με τη μέθοδο των ορθογωνίων σχημάτων μπορεί να είναι ανακριβής. . . . .	84
4.6 Παράδειγμα εικόνας για τη δημιουργία μάσκας της ελεύθερης θέσης στάθμευσης. . . . .	85
4.7 Δυνατές θέσεις εκκίνησης του αυτοκινήτου. . . . .	86
4.8 Παράδειγμα ανεπάρκειας των αισθητήρων. Παρατηρούμε ότι δεν ανισχνεύεται το ροζ αυτοκίνητο, παρόλο που βρίσκεται εντός του βεληνεκούς των ακτίνων. . . . .	91
 5.1 Εργαλεία εκπαίδευσης. . . . .	95
5.2 Πλήθος εκπαιδεύσεων ανά αλγόριθμο. . . . .	101
5.3 Μέση διάρκεια εκπαίδευσης ανά αλγόριθμο. . . . .	101
5.4 Προσδοκίες και πραγματικότητα στην εκπαίδευση πρακτόρων ενισχυτικής μάθησης (Amid 2018). . . . .	103
5.5 Δύο εκπαιδεύσεις του αλγορίθμου PPO: η εκπαίδευση 47 με entropy coefficient = 0.01 και η εκπαίδευση 47B με entropy coefficient = 0. . . . .	104
5.6 Παράδειγμα reward hacking (Raffin 2021). . . . .	105
5.7 Περίπτωση reward hacking στο περιβάλλον αυτόματης στάθμευσης. Ο πράκτορας έμαθε να πετυχαίνει μεγαλύτερη ανταμοιβή, χωρίς να παρκάρει. . . . .	106
5.8 Παράδειγμα σύγκλισης σε τοπικό μέγιστο. . . . .	107
5.9 Αρχιτεκτονική νευρωνικού δικτύου αλγορίθμων με διακριτό χώρο ενεργειών. . . . .	118
5.10 Αρχιτεκτονική νευρωνικού δικτύου αλγορίθμων με συνεχή χώρο ενεργειών. . . . .	120
5.11 Συνάρτηση με αραιές ανταμοιβές - Άμεση στάθμευση. . . . .	121
5.12 Συνάρτηση με διαμόρφωση ανταμοιβής - Άμεση στάθμευση. . . . .	123
5.13 Συνάρτηση με διαμόρφωση ανταμοιβής - Κανονική στάθμευση. . . . .	125
5.14 Τελική διακριτοποίηση χώρου καταστάσεων. . . . .	126
5.15 Εξασθένηση του $\epsilon$ . . . . .	127
5.16 Καλύτερη εκπαίδευση του Q-Learning στο επίπεδο δυσκολίας 1. . . . .	129
5.17 Καλύτερη εκπαίδευση του Q-Learning στο επίπεδο δυσκολίας 2. . . . .	130
5.18 Καλύτερη εκπαίδευση του Q-Learning στο επίπεδο δυσκολίας 3. . . . .	131
5.19 Καλύτερες εκπαιδεύσεις του αλγορίθμου PPO: με κόκκινο χρώμα στο επίπεδο μεταξύ 1 και 2, με γκρι χρώμα στο επίπεδο 2 και με μπλε χρώμα στο επίπεδο 3. . . . .	134
5.20 Καλύτερες εκπαιδεύσεις του αλγορίθμου SAC: με μπλε χρώμα στο επίπεδο 2, με ροζ χρώμα στο επίπεδο 3 και με κόκκινο χρώμα στο επίπεδο 4. . . . .	136
5.21 Καλύτερες εκπαιδεύσεις του ολγορίθμου TD3: με κόκκινο χρώμα στο επίπεδο 2, με τιρκουάζ χρώμα στο επίπεδο 3, με γκρι χρώμα στο επίπεδο 4 και με πορτοκαλί χρώμα στο επίπεδο 4 με αύξηση της τιμωρίας για τις συγκρούσεις. . . . .	138
5.22 Καλύτερες εκπαιδεύσεις του αλγορίθμου DDPG: με κόκκινο χρώμα στο επίπεδο 3, με ροζ χρώμα στο επίπεδο 3 με αύξηση της τιμωρίας για τις συγκρούσεις και με πορτοκαλί χρώμα στο επίπεδο 4. . . . .	139

6.1	Χρόνοι εκπαίδευσης των καλύτερων πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3. . . . .	142
6.2	Χρόνοι εκπαίδευσης των καλύτερων πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 4. . . . .	143
6.3	Ποσοστά επιτυχίας των πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3. . . . .	145
6.4	Μέσοι αριθμοί συγκρούσεων των πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3. . . . .	145
6.5	Μέσοι χρόνοι ολοκλήρωσης των πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3. . . . .	146
6.6	Τελικά σκορ των αλγορίθμων στο επίπεδο δυσκολίας 3. . . . .	147
6.7	Ποσοστά επιτυχίας των παικτών στο επίπεδο δυσκολίας 4. . . . .	148
6.8	Μέσοι αριθμοί συγκρούσεων των παικτών στο επίπεδο δυσκολίας 4. . . . .	148
6.9	Μέσοι χρόνοι στάθμευσης των παικτών στο επίπεδο δυσκολίας 4. . . . .	149
6.10	Τελικά σκορ των αλγορίθμων στο επίπεδο δυσκολίας 4, κανονικοποιημένα ως προς τις ανθρώπινες επιδόσεις. . . . .	150



# 1 Εισαγωγή

## 1.1 Στόχος

Η παρούσα εργασία πραγματεύεται την εκπαίδευση πρακτόρων τεχνητής νοημοσύνης, προκειμένου να μάθουν να επιτελούν μία εργασία (*task*), σε περιβάλλον παιχνιδιού. Η εκπαίδευση γίνεται χρησιμοποιώντας διαφορετικούς αλγορίθμους Μηχανικής Μάθησης (*Machine Learning*). Τελικός στόχος αποτελεί η σύγκριση των αλγορίθμων αυτών, η οποία πραγματοποείται σε 2 άξονες:

- όσον αφορά τον **χρόνο εκπαίδευσης**, δηλαδή τον χρόνο που απαιτήθηκε για να πετύχει ο πράκτορας την καλύτερη επίδοση του στο παιχνίδι και
- όσον αφορά την **τελική επίδοση**, δηλαδή το πόσο καλά επιτελεί ο εκπαιδευμένος πράκτορας την εργασία του στο παιχνίδι.

Αναλυτικότερα, το περιβάλλον παιχνιδιού που αναπτύχθηκε είναι ένας χώρος parking και η εργασία που ο πράκτορας καλείται να μάθει είναι η στάθμευση ενός αυτοκινήτου σε μία συγκεκριμένη, αλλά τυχαία θέση στον χώρο αυτό. Επιλέχθηκε η συγκεκριμένη εργασία μετά από αρκετή σκέψη, καθώς εκτός του ενδιαφέροντος που παρουσιάζει, προσφέρει και τη δυνατότητα μελέτης ενός προβλήματος με εφαρμογές στον πραγματικό κόσμο. Πράγματι, η στάθμευση αποτελεί ένα πρόβλημα που καλούνται να αντιμετωπίσουν καθημερινά τα αυτόνομα οχήματα, τα οποία πρέπει να είναι σε θέση να επιτελούν αυτή την εργασία με ασφάλεια, ταχύτητα και αποτελεσματικότητα. Επομένως, η επιτυχής επίλυση του προβλήματος της στάθμευσης μέσω της μηχανικής μάθησης, ακόμα και σε περιβάλλον προσομοίωσης, αποτελεί ένα πρώτο βήμα προς την κατεύθυνση της ανάπτυξης αυτόνομων οχημάτων.

Ακόμα, σε προσωπικό επίπεδο και ως αρχάριος σε θέματα τεχνητής νοημοσύνης και μηχανικής μάθησης, στόχος μου μέσα από την εκπόνηση της παρούσας εργασίας, αποτέλεσε η πρώτη εξοικείωση με το αντικείμενο και η κατανόηση των βασικών αρχών και τρόπων λειτουργίας συστημάτων τεχνητής νοημοσύνης. Έχοντας πλέον ολοκληρώσει την εργασία, θεωρώ πως ο στόχος αυτός επετεύχθη και μάλιστα ευχάριστα, με τον ψυχαγωγικό τρόπο που προσφέρει η ενασχόληση με ένα παιχνίδι.

Ωστόσο, κατά την εκπόνηση της παρούσας εργασίας, αντιμετώπισα αρκετά προβλήματα και υπέπεισα σε λάθη, τα οποία μου κόστισαν σε χρόνο και ενέργεια. Η αναζήτηση πληροφοριών στο

## 1.2 Διάρθρωση Εργασίας

---

διαδίκτυο είναι μία χρονοβόρα διαδικασία, εξαιτίας του τεράστιου όγκου διαθέσιμης πληροφορίας. Κάποιος άπειρος μπορεί εύκολα να χαθεί στην πληθώρα διαφορετικών πηγών και επιστημονικών δημοσιεύσεων και να σπαταλήσει χρόνο σε πληροφορίες μη χρήσιμες για αυτόν ή υπερβολικά λεπτομερείς.

Για αυτό τον λόγο, μέσα από το κείμενο αυτής της εργασίας, επιθυμώ να βοηθήσω νέους ερευνητές του πεδίου να εισαχθούν πιο ομαλά σε αυτό. Στο πνεύμα αυτό, θα προσπαθήσω να εξηγώ τα θέματα της παρούσας εργασίας με τρόπο απλό και σαφή, ώστε να μην χρειάζεται κάποιος να έχει προηγούμενη εμπειρία για να τα κατανοήσει. Έτσι, ελπίζω αυτή η διατριβή να χρησιμοποιηθεί ως αφετηρία από μελλοντικούς φοιτητές με παρόμιο αντικείμενο μελέτης και η αναγνώση της δικής μου εμπειρίας να τους αποτρέψει από την επανάληψη των ίδιων λαθών.

## 1.2 Διάρθρωση Εργασίας

Η παρούσα εργασία χωρίζεται σε 7 κεφάλαια, τα οποία αποσκοπούν στην εύκολη διάκριση και κατανόηση των επιμέρους στοιχείων της. Στη συνέχεια, παρουσιάζεται μία σύντομη περιγραφή του περιεχομένου του κάθε κεφαλαίου, πλην της εισαγωγής.

Στο **Κεφάλαιο 2** γίνεται μία συνοπτική παρουσίαση του πεδίου εφαρμογής της εργασίας, δηλαδή του χώρου των αυτόνομων οχημάτων και της αυτόματης στάθμευσης. Ακόμα, διεξάγεται μία βιβλιογραφική επισκόπηση σε προηγούμενες, σχετικές εργασίες.

Στο **Κεφάλαιο 3** διατυπώνεται το βασικό, θεωρητικό υπόβαθρο, που απαιτείται για την κατανόηση αυτής της εργασίας. Η ανάπτυξη της θεωρίας ξεκινά από εισαγωγικές έννοιες όπως η Τεχνητή Νοημοσύνη και η Μηχανική Μάθηση και καταλήγει στο κεντρικό, θεωρητικό αντικείμενο της εργασίας, την Ενισχυτική Μάθηση.

Στο **Κεφάλαιο 4** παρουσιάζεται το παιχνίδι στάθμευσης που κατασκευάστηκε, το οποίο λειτούργησε ως το περιβάλλον εκπαίδευσης των πρακτόρων.

Στο **Κεφάλαιο 5** αναλύεται η εκπαίδευση των πρακτόρων. Πρώτα περιγράφεται η διαδικασία της εκπαίδευσης και ο τρόπος υλοποίησης της. Στη συνέχεια, αναλύονται ορισμένα χρήσιμα συμπεράσματα της διαδικασίας αυτής, δηλαδή οι πρακτικές που εφαρμόστηκαν, για την αντιμετώπιση διαφόρων προβλημάτων που προέκυψαν. Τέλος, παρατίθενται τα αποτελέσματα από τις πιο επιτυχημένες εκπαίδευσεις του κάθε αλγορίθμου.

Στο **Κεφάλαιο 6** πραγματοποιείται η αξιολόγηση των αποτελεσμάτων της εκπαίδευσης και η σύγκριση των διαφορετικών αλγορίθμων. Έπειτα, τεκμηριώνονται τα τελικά πορίσματα της εργασίας.

Στο **Κεφάλαιο 7** προτείνονται ιδέες για τη μελλοντική βελτίωση και επέκταση της εργασίας.

## 2 Επισκόπηση του χώρου

Σε αυτό το κεφάλαιο, θα πραγματοποιήσουμε μία επισκόπηση του χώρου που επιλέχθηκε ως αντικείμενο μελέτης της παρούσας εργασίας, δηλαδή του χώρου της αυτόματης στάθμευσης και γενικότερα, της αυτόνομης οδήγησης. Αρχικά, στην Ενότητα 2.1, θα παρουσιάσουμε τον τρόπο λειτουργίας και την τρέχουσα κατάσταση της τεχνολογίας των αυτόνομων οχημάτων στον πραγματικό κόσμο. Στη συνέχεια, στην Ενότητα 2.2, θα αναφερθούμε σε προηγούμενες εργασίες και επιστημονικές δημοσιεύσεις που ασχολήθηκαν με την αυτόματη στάθμευση σε περιβάλλοντα προσομοιώσεων και παιχνιδιών.

### 2.1 Αυτόνομα οχήματα

#### 2.1.1 Ορισμός

Ως «αυτόνομο όχημα», ορίζεται η τεχνολογία για μερική ή πλήρη αντικατάσταση του ανθρώπινου οδηγού στην πλοήγηση ενός οχήματος από την αφετηρία στον προορισμό, αποφεύγοντας κινδύνους του δρόμου και ανταποκρινόμενη στις κυκλοφοριακές συνθήκες (University of Michigan Center for Sustainable Systems 2023). Τα αυτόνομα οχήματα χωρίζονται σε 6 επίπεδα αυτοματισμού, με βάση το βαθμό στον οποίο ένα όχημα μπορεί να λειτουργήσει χωρίς ανθρώπινη παρέμβαση, σύμφωνα με τον οργανισμό SAE International (Society of Automobile Engineers 2021). Τα 6 αυτά επίπεδα φαίνονται στην Εικόνα 2.1.

Τα πρώτα 3 επίπεδα (0-2) αναφέρονται σε οχήματα που απαιτούν την παρουσία ενός ανθρώπου οδηγού, αλλά χρησιμοποιούν αυτοματισμούς για την ασφάλεια, όπως προειδοποιήσεις για τυφλά σημεία και αυτόματο φρενάρισμα. Τα επίπεδα 3 και 4 αντιπροσωπεύουν τεχνολογία στην οποία το όχημα είναι αυτόνομο κάτω από συγκεκριμένες συνθήκες, ενώ κάτω από άλλες συνθήκες χρειάζεται η παρέμβαση ενός ανθρώπου. Τέλος, το επίπεδο 5 είναι το μόνο επίπεδο στο οποίο ένα όχημα θεωρείται πλήρως αυτόνομο και δεν απαιτείται ποτέ ανθρώπινη παρέμβαση.

#### 2.1.2 Αρχές λειτουργίας

Τα αυτόνομα οχήματα στηρίζονται κυρίως στην τεχνητή νοημοσύνη και απαιτούν μεγάλο πλήθος δεδομένων για την εκπαίδευση τους. Τα δεδομένα αυτά συλλέγονται από αισθητήρες, όπως κάμερες,

## 2.1 Αυτόνομα οχήματα

---

**SAE J3016™ LEVELS OF DRIVING AUTOMATION™**  
 Learn more here: [sae.org/standards/content/j3016\\_202104](https://sae.org/standards/content/j3016_202104)

Copyright © 2021 SAE International. The summary table may be freely copied and distributed AS-IS provided that SAE International is acknowledged as the source of the content.

	SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?	You <u>are driving</u> whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering	You <u>must constantly supervise</u> these support features; you must steer, brake or accelerate as needed to maintain safety		You <u>are not driving</u> when these automated driving features are engaged – even if you are seated in "the driver's seat"	When the feature requests, you must drive	These automated driving features will not require you to take over driving
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering <b>OR</b> brake/acceleration support to the driver	These features provide steering <b>AND</b> brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	<ul style="list-style-type: none"> <li>• automatic emergency braking</li> <li>• blind spot warning</li> <li>• lane departure warning</li> </ul>	<ul style="list-style-type: none"> <li>• lane centering <b>OR</b></li> <li>• adaptive cruise control</li> </ul>	<ul style="list-style-type: none"> <li>• lane centering <b>AND</b></li> <li>• adaptive cruise control at the same time</li> </ul>	<ul style="list-style-type: none"> <li>• traffic jam chauffeur</li> </ul>	<ul style="list-style-type: none"> <li>• local driverless taxi</li> <li>• pedals/steering wheel may or may not be installed</li> </ul>	<ul style="list-style-type: none"> <li>• same as level 4, but feature can drive everywhere in all conditions</li> </ul>

Copyright © 2021 SAE International.

	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering <b>OR</b> brake/acceleration support to the driver	These features provide steering <b>AND</b> brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	<ul style="list-style-type: none"> <li>• automatic emergency braking</li> <li>• blind spot warning</li> <li>• lane departure warning</li> </ul>	<ul style="list-style-type: none"> <li>• lane centering <b>OR</b></li> <li>• adaptive cruise control</li> </ul>	<ul style="list-style-type: none"> <li>• lane centering <b>AND</b></li> <li>• adaptive cruise control at the same time</li> </ul>	<ul style="list-style-type: none"> <li>• traffic jam chauffeur</li> </ul>	<ul style="list-style-type: none"> <li>• local driverless taxi</li> <li>• pedals/steering wheel may or may not be installed</li> </ul>	<ul style="list-style-type: none"> <li>• same as level 4, but feature can drive everywhere in all conditions</li> </ul>

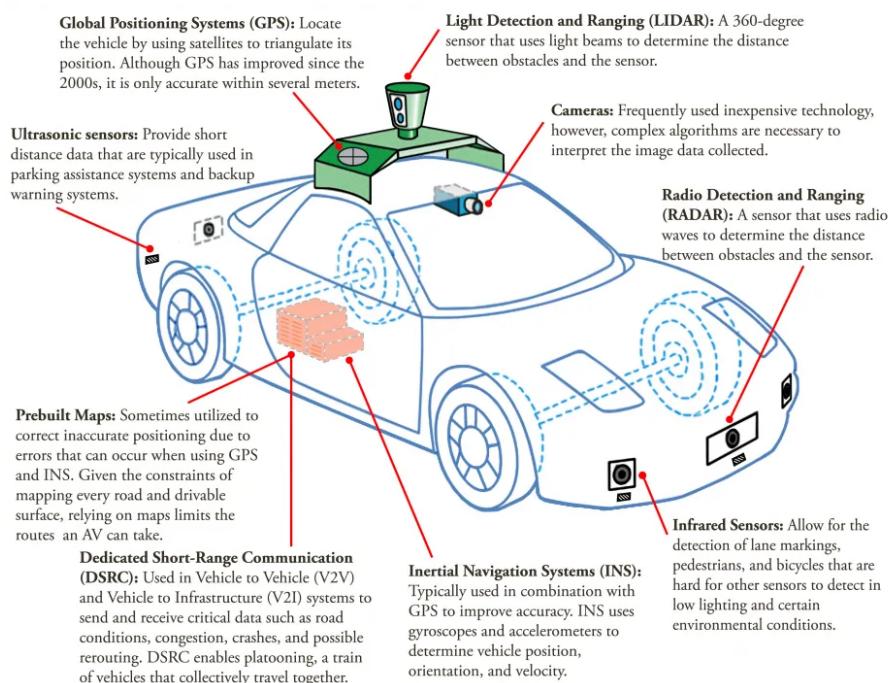
**Εικόνα 2.1.** Επίπεδα αυτοματισμού αυτόνομων οχημάτων (Society of Automobile Engineers 2021).

ραντάρ και λέιζερ (*LIDAR - Light Detection and Ranging*). Οι κάμερες χρησιμοποιούν Μηχανική Όραση (*Computer Vision*) για να αναγνωρίσουν τα αντικείμενα, ενώ τα ραντάρ και τα LIDAR χρησιμεύουν για την ανίχνευση της απόστασης και της ταχύτητας των αντικειμένων. Πιο συγκεκριμένα, η τεχνολογία LIDAR εκπέμπει μία ταχεία αλληλουχία από πολύ μικρούς παλμούς λέιζερ και μετρά το χρόνο που χρειάζεται για να επιστρέψουν από τα αντικείμενα που βρίσκονται στο δρόμο. Από το χρόνο αυτό, μπορεί να υπολογιστεί η απόσταση των αντικειμένων από το όχημα, ενώ από τη διαφορά των χρόνων μεταξύ διαδοχικών παλμών αντλούνται πληροφορίες για το σχήμα του κάθε αντικειμένου. Μάλιστα, χρησιμοποιώντας τεχνολογίες Ενσωματωμένης Φωτονικής (*Integrated Photonics*), όπως διαμορφωτές Mach-Zender και ανιχνευτές φωτός, η ακρίβεια στην ανάλυση των σχημάτων μπορεί να φτάσει το 1mm, κάτι που ξεπερνάει την ανθρώπινη όραση και αντίληψη (Saini 2019). Επομένως, με αυτούς τους τρόπους τα αυτόνομα οχήματα είναι σε θέση να εντοπίζουν και να αναγνωρίζουν τα στοιχεία του περιβάλλοντος οδήγησης, όπως φανάρια, δέντρα, πεζούς, πινακίδες κυκλοφορίας κ.ά. Στη συνέχεια, τα δεδομένα που συλλέγονται από τους αισθητήρες στέλνονται στο λογισμικό του οχήματος, όπου μέσω νευρωνικών δικτύων και αλγορίθμων μηχανικής μάθησης λαμβάνονται οι αποφάσεις για την κίνηση του.

Ακόμα, πολλές φορές χρησιμοποιείται η τεχνολογία *Geofencing* για να βοηθήσει στην πλοήγηση των αυτόνομων οχημάτων. Η τεχνολογία αυτή βασίζεται στο σύστημα Global Positioning System

(GPS) για να δημιουργήσει εικονικά όρια (geofences) σε μία συγκεκριμένη γεωγραφική περιοχή. Τα εικονικά όρια χρησιμοποιούνται για να να ενεργοποιήσουν αυτόματες ενέργειες ή ειδοποιήσεις όταν ένα οχημα εισέρχεται ή εξέρχεται από αυτά. Με αυτόν τον τρόπο, τα αυτόνομα οχήματα μπορούν να αναγνωρίσουν την περιοχή στην οποία κινούνται και να προσαρμόσουν την κίνησή τους ανάλογα (Gillis 2024).

Οι παραπάνω τεχνολογίες των αυτόνομων οχημάτων παρουσιάζονται στην Εικόνα 2.2.



**Εικόνα 2.2.** Τεχνολογίες αυτόνομων οχημάτων (University of Michigan Center for Sustainable Systems 2023)

### 2.1.3 Πλεονεκτήματα και Μειονεκτήματα

Η ευρεία υιοθέτηση των αυτόνομων οχημάτων θα έχει θετικές και αρνητικές επιπτώσεις στην κοινωνία, οι οποίες περιγράφονται αναλυτικά και με παραστατικό τρόπο στο (Bright Side 2020). Τα σημαντικότερα πλεονεκτήματα και μειονεκτήματα της χρήσης των αυτόνομων οχημάτων παρατίθενται παρακάτω.

Στα πλεονεκτήματα περιλαμβάνονται:

- η μείωση των τροχαίων ατυχημάτων. Σύμφωνα με το (Singh 2015), περισσότερο από 90% των τροχαίων ατυχημάτων προκαλούνται από τον ανθρώπινο παράγοντα όπως απόσπαση

## 2.1 Αυτόνομα οχήματα

---

προσοχής, κακή λήψη αποφάσεων ή κατανάλωση αλκοόλ. Επομένως, η αυτοματοποίηση της οδήγησης μπορεί να μειώσει σημαντικά τον αριθμό των ατυχημάτων.

- **η δυνατότητα μετακίνησης για άτομα που δεν είναι ικάνα να οδηγήσουν, όπως ηλικιωμένοι ή άτομα με αναπηρία.**
- **η μείωση της κυκλοφοριακής συμφόρησης.** Αυτό είναι εφικτό χάρη στην επικοινωνία μεταξύ των αυτόνομων οχημάτων και το συντονισμό τους.
- **η μείωση της περιβαλλοντικής ρύπανσης.** Τα αυτόνομα οχήματα πετυχαίνουν πιο αποδοτική οδήγηση, η οποία, σύμφωνα με μελέτη του πανεπιστημίου του Michigan (Erickson 2018), μπορεί να οδηγήσει σε μείωση των εκπομπών ρύπων κατά 9% σε σύγκριση με τα συμβατικά οχήματα.
- **η αύξηση του ελεύθερου χρόνου των ανθρώπων, καθώς δεν θα χρειάζεται να έχουν την προσοχή τους στην οδήγηση.**
- **η διευκόλυνση της στάθμευσης.** Τα αυτόνομα οχήματα μπορούν να αφήνουν τους επιβάτες τους απευθείας στον προορισμό τους κι έπειτα να αναζητούν χώρο στάθμευσης και να παρκάρουν αυτόματα.

Στα μειονεκτήματα περιλαμβάνονται:

- **η μείωση θέσεων εργασίας.** Η ευρεία χρήση των αυτόνομων οχημάτων ενδέχεται να επηρεάσει εργαζόμενους στον τομέα της οδήγησης όπως οδηγούς ταξί και φορτηγών.
- **η αύξηση του κόστους.** Η απαραίτητη τεχνολογία σε υλικό και σε λογισμικό για τη λειτουργία των αυτόνομων οχημάτων μπορεί να αυξήσει σημαντικά το κόστος τους σε σχέση με τα συμβατικά οχήματα.
- **η έλλειψη νομοθεσίας.** Η ανάπτυξη των αυτόνομων οχημάτων αντιμετωπίζει πληθώρα νομικών προβλημάτων που προκύπτουν από τη χρήση τους. Για παράδειγμα, ένα ερώτημα που πρέπει να απαντηθεί, είναι το ποιος είναι υπεύθυνος σε περίπτωση ατυχήματος: Ο ιδιοκτήτης του οχήματος, ο κατασκευαστής του ή ο προγραμματιστής του λογισμικού;
- **το ηλεκτρονικό έγκλημα.** Το λογισμικό των αυτόνομων οχημάτων θα αποτελέσει στόχο κακοβουλών προγραμματιστών (hackers) που θα επιδιώκουν να κλέψουν προσωπικά δεδομένα των χρηστών ή ακόμα και να ανακατευθύνουν τα οχήματα ή να προκαλέσουν ατυχήματα .

### 2.1.4 Η κατάσταση σήμερα

Παρόλο που πολλοί ειδικοί του χώρου προέβλεπαν ότι τα αυτόνομα οχήματα θα κυκλοφορούσαν ήδη στους δρόμους, όπως ο ιδρυτής της Tesla Elon Musk, ο οποίος υποστήριζε πως από το 2020 τα αυτοκίνητα της εταιρίας του θα ήταν πλήρως αυτόνομα και δεν θα απαιτούσαν την προσοχή των επιβατών τους (Trudell 2024), η πραγματικότητα είναι διαφορετική.

Σήμερα, τα συστήματα αυτόματης οδήγησης προς πώληση ανήκουν στο επίπεδο 2, όπως το Super Cruise της εταιρίας Grand Motors και το Full Self Driving της εταιρίας Tesla. Τα συστήματα αυτά

έχουν χαρακτηριστικά όπως το σύστημα προσαρμοστικού ρυθμού (*Adaptive Cruise Control*) για την αυτόματη ρύθμιση της ταχύτητας, το σύστημα παρακολούθησης λωρίδας (*Lane Keeping*) για τη διατήρηση του οχήματος στη λωρίδα του και το σύστημα αυτόματης στάθμευσης (*Autopark*) για τη μετακίνηση σε παράλληλες ή κάθετες θέσεις στάθμευσης (Tucker 2024). Ωστόσο, εξακολουθεί να είναι απαραίτητη η συνεχής προσοχή και ετοιμότητα του οδηγού.

Υπάρχουν ακόμα πολλά εμπόδια που καθυστερούν την ανάπτυξη και εμπορική χρήση πλήρως αυτόνομων οχημάτων. Κάποια από αυτά αναφέρθηκαν νωρίτερα στα μειονεκτήματα τους, όπως η ανάγκη θέσπισης σχετικής νομοθεσίας, η διασφάλιση της ασφάλειας των δεδομένων και η ζητούμενη μείωση του κόστους παραγωγής των τεχνολογιών αυτόνομης οδήγησης. Επίσης, πρόκληση αποτελεί το γεγονός ότι τα αυτόνομα οχήματα δεν μπορούν να λειτουργήσουν κάτω από κακές καιρικές συνθήκες, καθώς οι αισθητήρες τους εμποδίζονται από τη βροχή, το χιόνι ή την ομίχλη. Επιπλέον, η κακή κατάσταση των δρόμων, με παραδείγματα όπως λακούβες και ελλιπή σήμανση, δυσκολεύει την κατανόηση του περιβάλλοντος οδήγησης από τα αυτόνομα οχήματα. Όλοι οι παραπάνω περιορισμοί πρέπει να αρθούν ώστε τα αυτόνομα οχήματα να γίνουν πραγματικά ασφαλή, να κερδίσουν την εμπιστοσύνη των καταναλωτών και να επιτύχουν ευρεία υιοθέτηση.

Δυστυχώς, μεγάλη μερίδα του κόσμου δεν καταλαβαίνει ότι τα σημερινά συστήματα παρέχουν μόνο μερικώς αυτόματη οδήγηση. Αυτό οφείλεται κυρίως στην παραπλανητική προώθηση (*marketing*) από εταιρίες όπως η Tesla, η οποία με τον όρο «*Full Self Driving*» δημιουργεί την εντύπωση ότι τα αυτοκίνητά της είναι πλήρως αυτόνομα. Αυτό έχει ως αποτέλεσμα την υπερβολική χαλάρωση των οδηγών και την αύξηση των ατυχημάτων. Σύμφωνα με δεδομένα της Εθνική Υπηρεσία Οδικής Ασφάλειας των ΗΠΑ (NHTSA), από το 2019 έχουν διερευνηθεί 736 ατυχήματα που συνέβησαν σε αυτοκίνητα της Tesla που ήταν σε λειτουργία αυτόματης οδήγησης, ενώ από αυτά προέκυψαν 17 θάνατοι (Blanco 2023).

Πλέον, οι ειδικοί είναι πιο μετριοπαθείς στις προβλέψεις τους, εκτιμώντας πως θα χρειαστούν ακόμα αρκετές δεκαετίες για να επιτευχθεί το επίπεδο 5, δηλαδή το πλήρως αυτόνομο όχημα. Σε μία έρευνα του 2023 από τη διεθνή εταιρεία παροχής συμβουλευτικών υπηρεσιών McKinsey and Company (Deichmann κ.ά. 2023), προβλέπεται ότι το 2030, το 12% των νέων επιβατικών αυτοκινήτων θα πωλούνται με τεχνολογίες αυτόνομης οδήγησης επιπέδου 3 ή μεγαλύτερου, ενώ το 2035 το 37% αυτών θα έχουν προηγμένες τεχνολογίες αυτόνομης οδήγησης.

## 2.2 Προηγούμενη έρευνα

Στον τομέα της τεχνητής νοημοσύνης των αυτόνομων οχημάτων, έχουν πραγματοποιηθεί πολλές εργασίες και έρευνες τα τελευταία χρόνια, σε περιβάλλοντα προσομοιώσεων. Οι περισσότερες από αυτές, εστιάζουν στο κομμάτι της οδήγησης, όπως η εργασία των (Anzalone κ.ά. 2022), στην οποία χρησιμοποιείται ο αλγόριθμος PPO και η τεχνική της Κλιμακωτής Μάθησης (*Curriculum Learning*),

## 2.2 Προηγούμενη έρευνα

---

για την εκπαίδευση ενός πράκτορας τεχνητής νοημοσύνης σε σενάρια αυτόματης οδήγησης. Άλλο παράδειγμα αποτελεί το άρθρο των (Loiacono κ.ά. 2010), στο οποίο χρησιμοποιείται ο αλγόριθμος Q-Learning για την εκπαίδευση ενός πράκτορα στην προσπέραση άλλων οχημάτων.

Συχνά, τα τελικά αποτελέσματα τέτοιου είδους εργασίων καταγράφονται σε video και αναρτώνται σε πλατφόρμες όπως το YouTube. Αυτό συνηθίζεται, καθώς η κίνηση του πράκτορα δεν μπορεί να αναπαρασταθεί σε ένα επιστημονικό άρθρο, ενώ το video παρέχει μια πιο παραστατική εικόνα της λειτουργίας του αλγορίθμου στην πράξη. Έτσι, γίνεται ευκολότερα κατανοητό στον άνθρωπο το επίπεδο οδήγησης που έχει επιτύχει ο πράκτορας. Επίσης, τα video αυτού του είδους είναι μία καλή εισαγωγή στον χώρο της Ενισχυτικής Μάθησης, επειδή εξηγούν με σύντομο και απλό τρόπο τις βασικές αρχές του πεδίου και τη διαδικασία εκπαίδευσης του πράκτορα. Για αυτό, προτείνω στους αρχάριους του αντικειμένου να παρακολουθήσουν κάποια από αυτά τα video, προτού ασχοληθούν με την ανάγνωση τεχνικών άρθρων. Κάποια αξιοσημείωτα video είναι τα (Porro 2020), (NeuralNine 2021), (Saravia 2020), στα οποία χρησιμοποιείται ο γενετικός αλγόριθμος NEAT και τα (Code-Bullet 2019), (AI-Forge 2022), στα οποία χρησιμοποιούνται οι αλγόριθμοι Q-Learning και PPO αντίστοιχα. Στα παραπάνω video, πρώτα δημιουργείται ένα παιχνίδι αγώνων ταχύτητας (*car racing game*), στο οποίο ο πράκτορας εκπαιδεύεται να ολοκληρώνει τον γύρο της πίστας, όσο το δυνατόν πιο γρήγορα και χωρίς να παρεκτρέπεται εκτός δρόμου. Μετά από αρκετά βήματα εκπαίδευσης, οι πράκτορες φτάνουν σε ικανοποιητικό επίπεδο οδήγησης.

Παράλληλα, υπάρχουν και εργασίες που επικεντρώνονται στο κομμάτι της αυτόματης στάθμευσης (*Car Parking*), αν και είναι πολύ λιγότερες σε σχέση με τις εργασίες οδήγησης. Για παράδειγμα, η εργασία (Lai 2018) χρησιμοποιεί τον αλγόριθμο Q-Learning για την εκπαίδευση ενός πράκτορα στη στάθμευση σε παράλληλη θέση. Ενδιαφέρουσα αποτελεί επίσης η εργασία των (Mujumdar, Shah, και Cui 2020), στην οποία ο πράκτορας καλείται να φτάσει σε μία από τις διαθέσιμες θέσεις στάθμευσης, χωρίς να συγκρουστεί με άλλα οχήματα ή πεζούς και χρησιμοποιείται ο αλγόριθμος PPO καθώς και ο αλγόριθμος μίμησης GAIL. Ακόμα, η εργασία των (Moreira 2021) διεξάγει σύγκριση των αλγορίθμων DDPG, SAC και TD3 για την εργασία της αυτόματης στάθμευσης. Τέλος, ξεχωριστή αναφορά αξιζει το video του (Arzt 2019), καθώς ήταν αυτό που μου κίνησε πρώτο το ενδιαφέρον και με ενέπνευσε να ασχοληθώ με το θέμα της αυτόματης στάθμευσης.

# 3 Βασική Θεωρία

## 3.1 Τεχνητή Νοημοσύνη

### 3.1.1 Ορισμός

Οι Stuart Russell και Peter Norvig, στο βιβλίο τους “Artificial Intelligence: A Modern Approach” ορίζουν την τεχνητή νοημοσύνη ως «το επιστημονικό πεδίο το οποίο αναπτύσσει και μελετά λογισμικό και μεθόδους που επιτρέπουν στις μηχανές να αντιλαμβάνονται το περιβάλλον τους και να χρησιμοποιούν τη μάθηση και τη νοημοσύνη για να επιλέξουν δράσεις που μεγιστοποιούν τις πιθανότητες επίτευξης καθορισμένων στόχων» (Russell και Norvig 2021). Με άλλα λόγια, η τεχνητή νοημοσύνη είναι η τεχνολογία που δίνει τη δυνατότητα στους υπολογιστές και τις μηχανές να προσομοιώνουν την ανθρώπινη νοημοσύνη και να αποκτούν δυνατότητες επίλυσης προβλημάτων. Το ευρύ πεδίο της τεχνητής νοημοσύνης περιλαμβάνει πολλές διαφορετικές επιστήμες, από την Επιστήμη των Υπολογιστών (Computer Science), την Ανάλυση Δεδομένων (Data Analytics), το Υλικό (Hardware), τη Νευροεπιστήμη (Neuroscience), έως τη Γλωσσολογία (Linguistics) και τη Φιλοσοφία (Philosophy).

### 3.1.2 Μοντέλα και Πράκτορες τεχνητής νοημοσύνης

Το τελικό αποτέλεσμα μίας εφαρμογής τεχνητής νοημοσύνης συχνά καλείται μοντέλο (*model*) ή πράκτορας (*agent*), με τους όρους αυτούς να χρησιμοποιούνται εναλλάξιμα. Ωστόσο, υπάρχει λεπτή διάκριση μεταξύ τους, καθώς έχουν μοναδικές λειτουργίες κι εξυπηρετούν διαφορετικούς σκοπούς (Ezeokeke 2024).

Συγκεκριμένα, τα μοντέλα τεχνητής αναφέρονται σε συστήματα που εκπαιδεύονται σε συγκεκριμένα σύνολα δεδομένων και ως εκ τούτου, είναι συνήθως περιορισμένα στην εκτέλεση συγκεκριμένων εργασιών. Αποτελούν τον “εγκέφαλο” πολλών εφαρμογών τεχνητής νοημοσύνης, όπως τα μοντέλα αναγνώρισης εικόνων για την ανίχνευση προσώπων ή τα γλωσσικά μοντέλα που χρησιμοποιούνται στη μετάφραση κειμένου.

Αντίθετα, οι πράκτορες τεχνητής νοημοσύνης είναι πιο πολύπλοκες οντότητες που όχι μόνο χρησιμοποιούν μοντέλα τεχνητής νοημοσύνης, αλλά αλληλεπιδρούν με το περιβάλλον τους για

### 3.1 Τεχνητή Νοημοσύνη

---

να επιτύχουν τους στόχους τους. Συνεπώς, πρόκειται για συστήματα που χαρακτηρίζονται από αυτονομία και είναι σε θέση να ενεργούν ανεξάρτητα. Χρησιμοποιούνται σε εφαρμογές που απαιτείται η λήψη αποφάσεων σε πραγματικό χρόνο, όπως οι εικονικοί βοηθοί, τα αυτόνομα οχήματα και οι ρομποτικές εφαρμογές.

Επομένως, τα μοντέλα τεχνητής νοημοσύνης μπορούν να παρομοιαστούν με μία ισχυρή μηχανή, η οποία όμως χρειάζεται έναν ικανό χειριστή, τον πράκτορα, για να μετατρέψει την πληροφορία της σε δράση.

#### 3.1.3 Ιστορική Εξέλιξη

Ο Άλαν Τούρινγκ ήταν ο πρώτος που διεξήγαγε σημαντικές έρευνες στον τομέα που αυτός ονόμασε «Μηχανική νοημοσύνη» (*Machine Intelligence*). Το 1950, ο Τούρινγκ δημοσιεύει το άρθρο «Υπολογιστικά Μηχανήματα και νοημοσύνη», στο οποίο θέτει το ερώτημα: «Μπορούν οι μηχανές να σκέφτονται;» (Turing 1950). Για να απαντήσει αυτό το ερώτημα, προτείνει ένα τεστ, γνωστό ως «Turing Test». Το τεστ αυτό περιλαμβάνει έναν ανθρώπινο ανακριτή που κάνει ερωτήσεις σε δύο συμμετέχοντες, έναν άνθρωπο και έναν υπολογιστή, χωρίς να γνωρίζει κάθε φορά ποιός απαντάει. Ο στόχος είναι μέσα από τις απαντήσεις τους και εντός ενός χρονικού πλαισίου, ο ανακριτής να μπορέσει να διακρίνει ποιος είναι ο άνθρωπος και ποιος ο υπολογιστής. Έτσι, η ικανότητα του υπολογιστή να σκέπτεται προκύπτει από την πιθανότητα να τον ανογνωρίσει εσφαλμένα ο ανακριτής ως άνθρωπο. Μάλιστα, ο Τούρινγκ υποστηρίζει πως εάν ο υπολογιστής αντιδρά και συμπεριφερέται σαν νοήμων ον, τότε έχει συνείδηση. Αυτό το τεστ αποτελεί σημαντικό μέρος της ιστορίας της τεχνητής νοημοσύνης, καθώς και φιλοσοφική συζήτηση, η οποία επανήλθε στο προσκήνιο το 2022, με την κυκλοφορία του ChatGPT, με μερικούς να υποστηρίζουν πως πλέον έχουν επιτευχθεί τα κριτήρια του Turing Test.

Ο όρος «τεχνητή νοημοσύνη» χρησιμοποιήθηκε για πρώτη φορά το 1956 από τον Τζον Μακάρθι, ο οποίος αναγνωρίζεται ευρέως ως ο πατέρας της τεχνητής νοημοσύνης, λόγω της προσφοράς του στο αντικείμενο. Τότε, στο πρώτο συνέδριο τεχνητής νοημοσύνης στο Πανεπιστήμιο Dartmouth, ο Τζον Μακάρθι όρισε την τεχνητή νοημοσύνη ως την «επιστήμη και μεθοδολογία της δημιουργίας νοημόνων μηχανών» (McCarthy 2004).

Την δεκαετία του 1980, τα νευρωνικά δίκτυα που χρησιμοποιούν τον αλγόριθμο της οπισθοδιάδοσης (*backpropagation*) για την εκπαίδευσή τους, ξεκινάνε να χρησιμοποιούνται ευρέως σε εφαρμογές τεχνητής νοημοσύνης. Η τεχνολογία αυτή αποτελεί τη βάση πολλών συστημάτων τεχνητής νοημοσύνης που χρησιμοποιούνται σήμερα.

Η δεκαετία του 1990 σηματοδοτεί την έναρξη της εποχής της επιτυχίας των συστημάτων τεχνητής νοημοσύνης στην επίλυση προβλημάτων που απαιτούν ανθρώπινη νοημοσύνη. Αρχικά, το 1995, οι Stuart Russell και Peter Norvig δημοσιεύουν το βιβλίο «Artificial Intelligence: A Modern Approach»,

το οποίο αποτελεί ακόμα και σήμερα, το δημοφιλέστερο εγχειρίδιο της τεχνητής νοημοσύνης παγκοσμίως. Το 1997, ο υπολογιστής Deep Blue της IBM κερδίζει τον τότε παγκόσμιο πρωταθλητή στο σκάκι, Garry Kasparov. Αυτή ήταν η πρώτη φορά που ο τρέχων παγκόσμιος πρωταθλητής ηττήθηκε από έναν υπολογιστή.

Στα πιο σύγχρονα χρόνια, το ενδιαφέρον και η χρηματοδότηση στο πεδίο της τεχνητής νοημοσύνης αυξήθηκαν σημαντικά μετά το 2012, όταν η αύξηση της υπολογιστικής ισχύος οδήγησε σε σημαντικές επιτυχίες στον τομέα της Βαθιάς Μάθησης (*Deep Learning*). Έτσι, το 2015, ο υπερυπολογιστής Minwa της Baidu χρησιμοποιεί ένα ειδικό βαθύ νευρωνικό δίκτυο (*Deep Neural Network*), γνωστό ως Συνελικτικό Νευρωνικό Δίκτυο (*Convolutional Neural Network*), για την αναγνώριση και κατηγοριοποίηση εικόνων και πετυχαίνει υψηλότερο βαθμό ακρίβειας από τον μέσο ανθρώπο. Το 2016, το πρόγραμμα AlphaGo της DeepMind -που επίσης τροφοδοτείται από ένα βαθύ νευρωνικό δίκτυο- κερδίζει τον παγκόσμιο πρωταθλητή στο παιχνίδι στρατηγικής Go. Η νίκη αυτή είναι εξίσου σημαντική με αυτήν έναντια στον Kasparov, δεδομένου του τεράστιου αριθμού πιθανών κινήσεων που υπάρχουν στο Go (πάνω από 14,5 τρισεκατομμύρια μετά από μόλις τέσσερις κινήσεις). Το 2023, η εμφάνιση των Μεγάλων Γλωσσικών Μοντέλων (*Large Language Models*), όπως το ChatGPT, προκαλεί μια τεράστια αύξηση στις επιδόσεις της τεχνητής νοημοσύνης και την αξία που έχει, για ανθρώπους και επιχειρήσεις.

Στη σημερινή εποχή, η παραγωγική τεχνητή νοημοσύνη (*Generative AI*) μπορεί να μάθει και να συνθέσει όχι μόνο ανθρώπινη γλώσσα, αλλά και άλλους τύπους δεδομένων, συμπεριλαμβανομένων των εικόνων, του βίντεο, της μουσικής, του κώδικα λογισμικού και ακόμη και των μοριακών δομών. Έτσι, το πεδίο της τεχνητής νοημοσύνης βρίσκεται σήμερα σε άνθιση, με εταιρείες όπως η OpenAI, η Google και η NVIDIA να αναπτύσσουν συνεχώς νέες, πρωτοποριακές εφαρμογές και υπηρεσίες που βασίζονται σε αυτήν.

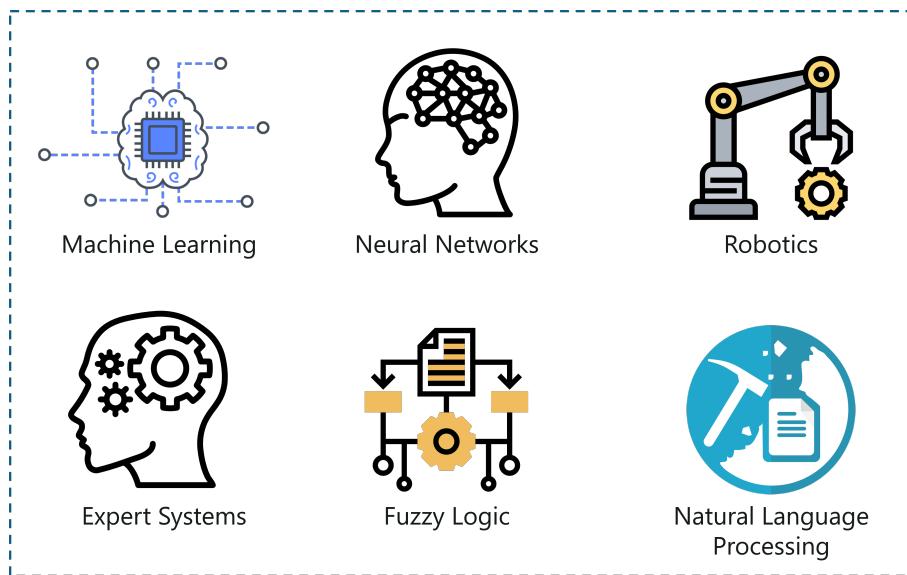
### 3.1.4 Κατηγορίες

Η τεχνητή νοημοσύνη αποτελεί ένα ευρύ πεδίο, το οποίο περιλαμβάνει πολλές διαφορετικές τεχνολογίες και εφαρμογές. Οι κύριοι κλάδοι της τεχνητής νοημοσύνης παρουσιάζονται στην *Εικόνα 3.1* και περιγράφονται συνοπτικά παρακάτω.

#### Μηχανική μάθηση

Αποτελεί τον κλάδο της τεχνητής νοημοσύνης που εστιάζει στην ανάπτυξη αλγορίθμων και μοντέλων που επιτρέπουν στους υπολογιστές να μαθαίνουν από μεγάλα σύνολα δεδομένων και να κάνουν προβλέψεις ή να παίρνουν αποφάσεις. Τα μοντέλα μηχανικής μάθησης εκπαιδεύονται και βελτιώνουν την απόδοση τους με το χρόνο, χωρίς να χρειάζεται να προγραμματίζονται ρητά για κάθε συγκεκριμένη εργασία. Η μηχανική μάθηση αναλύεται εκτενέστερα στην *Ενότητα 3.2*.

### 3.1 Τεχνητή Νοημοσύνη



Εικόνα 3.1. Κατηγορίες τεχνητής νοημοσύνης (Lateef 2024).

#### Νευρωνικά Δίκτυα (Βαθιά Μάθηση)

Η Βαθιά μάθηση είναι ένας υποκλάδος της μηχανικής μάθησης που χρησιμοποιεί νευρωνικά δίκτυα πολλών επιπέδων -εξού και το βαθιά-, για να προσομοιώσει την πολύπλοκη διαδικασία λήψης αποφάσεων του ανθρώπινου εγκεφάλου. Επομένως, η κύρια διαφορά της βαθιάς μάθησης από την παραδοσιακή μηχανική μάθηση είναι η δομή του νευρωνικού δικτύου. Τα μοντέλα παραδοσιακής μηχανικής μάθησης χρησιμοποιούν απλά νευρωνικά δίκτυα με έως ένα κρυφό επίπεδο. Αντίθετα, τα μοντέλα βαθιάς μάθησης χρησιμοποιούν δύο ή περισσότερα κρυφά επίπεδα (συχνά εκατοντάδες ή χιλιάδες) για να εκπαιδεύσουν τα μοντέλα τους (Johnson 2020). Η εκπαίδευση αυτή απαιτεί μεγάλα σύνολα δεδομένων κι επομένως, σημαντικούς υπολογιστικούς πόρους. Ωστόσο, τα μοντέλα της βαθιάς μάθησης μπορούν να αναγνωρίσουν πολύπλοκα μοτίβα σε δεδομένα όπως είκονες, κείμενα και ήχους, προκειμένου να παράγουν ακριβείς αναλύσεις και προβλέψεις. Έτσι, επιτυγχάνουν να λύσουν πιο πολύπλοκα προβλήματα, όπως η αναγνώριση εικόνων, η μετάφραση γλώσσας και η αυτόνομη οδήγηση. Συνολικά, η βαθιά μάθηση είναι η τεχνολογία που κινητοποιεί τις περισσότερες εφαρμογές της τεχνητής νοημοσύνης στη σύγχρονη εποχή.

#### Ρομποτική

Η ρομποτική αποτελεί τον κλάδο που επικεντρώνεται στη δημιουργία ευφυών μηχανών (*robot*), ικανών να εκτελούν εργασίες στον φυσικό κόσμο. Ενσωματώνει τεχνητής νοημοσύνης για να επιτρέψει στα ρομπότ να αντιλαμβάνονται το περιβάλλον τους, να λαμβάνουν αποφάσεις και να προβαίνουν σε δράσεις αυτόνομα. Έτσι, τα ρομπότ αποτελούν πράκτορες τεχνητής νοημοσύνης,

οι οποίοι αλληλεπιδρούν και προσαρμόζονται στις δυναμικές συνθήκες του πραγματικού κόσμου. Χαρακτηριστικό παράδειγμα τέτοιου ρομπότ αποτελεί το ανθρωποειδές Σοφία, η οποία διακρίνεται για τη ρεαλιστική ανθρώπινη εμφάνισή της, τα εκφραστικά χαρακτηριστικά του προσώπου της και τη δυνατότητα της να αλληλεπιδράσει και να συνομιλήσει με ανθρώπους (Hanson Robotics 2016).

### **Ειδικά συστήματα**

Το πεδίο των ειδικών συστημάτων της τεχνητής νοημοσύνης περιλαμβάνει την ανάπτυξη υπολογιστικών συστημάτων, που μιμούνται την ικανότητα λήψης αποφάσεων ενός ανθρώπου, ειδικού, σε συγκεκριμένο τομέα. Τέτοια συστήματα χρησιμοποιούν μια βάση γνώσης από γεγονότα και κανόνες (λογική if-then-else), μαζί με ένα μηχανισμό για να εφαρμόσουν τη γνώση αυτή σε νέες καταστάσεις και να επιλύσουν πολύπλοκα προβλήματα. Τα ειδικά συστήματα χρησιμοποιούνται ευρέως σε τομείς όπως η ιατρική διάγνωση, ο χρηματοοικονομικός σχεδιασμός και η ανίχνευση κακόβουλου λογισμικού.

### **Ασαφής λογική**

Η ασαφής λογική (*Fuzzy Logic*) είναι το πεδίο της τεχνητής νοημοσύνης που επιτρέπει στα συστήματα να χειρίστούν την αβεβαιότητα και την ασάφεια των δεδομένων. Αυτό επιτυγχάνεται χρησιμοποιώντας βαθμούς αληθείας (*degrees of truth*) έναντι αυστηρών τιμών (αληθής/ψευδής) της δυαδικής λογικής (*binary logic*). Έτσι, παρέχει προσαρμοστικότητα που επιτρέπει την μοντελοποίηση περίπλοκων, πραγματικών καταστάσεων. Η ασαφής λογική χρησιμοποιείται σε εφαρμογές που διαχειρίζονται ατελή δεδομένα, όπως τα συστήματα ελέγχου (π.χ. η ρύθμιση της θερμοκρασίας σε συστήματα κλιματισμού) και η λήψη αποφάσεων (π.χ. η πρόταση ιατρικών διαγνώσεων).

### **Επεξεργασία φυσικής γλώσσας**

Η επεξεργασία φυσικής γλώσσας (*Natural Language Processing - NLP*) είναι το υποπεδίο της τεχνητής νοημοσύνης που πραγματεύεται την ανάπτυξη τεχνολογιών που επιτρέπουν στους υπολογιστές να αναγνωρίζουν, να κατανοούν και να παράγουν ανθρώπινη γλώσσα. Για να το πετύχει αυτό, συνδυάζει την Υπολογιστική Γλωσσολογία (*Computational Linguistics*) με τη στατιστική μοντελοποίηση, τη μηχανική μάθηση και τη βαθιά μάθηση. Η έρευνα στην επεξεργασία φυσικής γλώσσας έχει επιφέρει την εποχή της παραγωγικής τεχνητής νοημοσύνης, με την κυκλοφορία των μεγάλων γλωσσικών μοντέλων, όπως το ChatGPT, που μπορούν να μάθουν και να συνθέσουν ανθρώπινη γλώσσα, αλλά και άλλους τύπους δεδομένων όπως εικόνες και βίντεο. Έτσι, η επεξεργασία φυσικής γλώσσας χρησιμοποιείται σε πολλές εφαρμογές, όπως οι ψηφιακοί βοηθοί, οι μηχανές μετάφρασης, οι μηχανές αναζήτησης και τα συστήματα αναγνώρισης φωνής.

### 3.1.5 Εφαρμογές

Σήμερα, η τεχνητή νοημοσύνη είναι το επίκεντρο πολλών τεχνολογιών που υποστηρίζουν υπηρεσίες και αγαθά, τα οποία χρησιμοποιούμε καθημερινά. Ορισμένες από τις πιο συνηθισμένες εφαρμογές της τεχνητής νοημοσύνης περιλαμβάνουν:

- **αναγνώριση ομιλίας:** η αυτόματη μετατροπή της προφορικής ομιλίας σε γραπτό κείμενο γίνεται με τεχνικές της επεξεργασίας φυσικής γλώσσας. Χρησιμοποιείται από ψηφιακούς βοηθούς όπως η Siri της Microsoft και η Alexa της Amazon.
- **αναγνώριση εικόνας:** η αναγνώριση και κατηγοριοποίηση εικόνων γίνεται από αλγόριθμους βαθιάς μάθησης, όπως τα συνελικτικά νευρωνικά δίκτυα και έχει εφαρμογές σε τομείς όπως η αυτόνομη οδήγηση και η αναγνώριση προσώπων.
- **αυτόματη μετάφραση:** εφαρμογές όπως το Google Translate χρησιμοποιούν αλγορίθμους βαθιάς μάθησης για να μεταφράσουν κείμενο από μία γλώσσα σε άλλη.
- **συστήματα προτάσεων:** τα συστήματα αυτά χρησιμοποιούν αλγορίθμους μηχανικής μάθησης για να αναλύσουν τα δεδομένα από πλατφόρμες όπως το YouTube, το Spotify και το Netflix και να προτείνουν περιεχόμενο στους χρήστες, με βάση τις προηγούμενες προτιμήσεις τους.
- **εμπορική προώθηση:** η τεχνητή νοημοσύνη χρησιμοποιείται στην ανάλυση δεδομένων από πλατφόρμες κοινωνικών δικτύων και ηλεκτρονικών καταστημάτων, όπως το Facebook και η Amazon, για την πρόβλεψη των προτιμήσεων των καταναλωτών και την προσωποποίηση των διαφημίσεων και των προσφορών.
- **συστήματα αναζήτησης:** οι αλγόριθμοι αναζήτησης όπως το Google Search χρησιμοποιούν τεχνολογίες τεχνητής νοημοσύνης για να παρέχουν ακριβέστερα αποτελέσματα αναζήτησης.
- **αυτόνομη οδήγηση:** τα αυτόνομα οχήματα χρησιμοποιούν μηχανική όραση για την αναγνώριση του περιβάλλοντος και βαθιά νευρωνικά δίκτυα για τη λήψη αποφάσεων κατά την οδήγηση.
- **οικονομική διαχείριση:** αλγόριθμοι τεχνητής νοημοσύνης για την ανάλυση δεδομένων χρησιμοποιούνται στην πρόβλεψη των χρηματιστηριακών αγορών και τη διαχείριση των επενδύσεων. Έτσι, επενδυτικές πλατφόρμες όπως η Betterment προσφέρουν αυτόματες, εξατομικευμένες οικονομικές συμβουλές, βασισμένες σε αλγόριθμους μηχανικής μάθησης.
- **παραγωγή περιεχομένου:** η παραγωγή περιεχομένου από αλγόριθμους βαθιάς μάθησης, όπως το ChatGPT για κείμενο, το DALL-E για εικόνες και το MuseNet για μουσική, επιτρέπει τη δημιουργία πολυμέσων με βάση την υπόδειξη (*prompt*) του χρήστη.
- **εξυπηρέτηση πελατών:** τα ρομπότ συνομιλίας (*chatbots*) χρησιμοποιούν τεχνολογίες επεξεργασίας φυσικής γλώσσας για να παρέχουν πληροφορίες στους χρήστες και να απαντούν στις ερωτήσεις τους. Μάλιστα, χρησιμοποιούν αλγορίθμους μηχανικής μάθησης

ώστε να βελτιώνουν την απόδοσή τους μέσω της εμπερίας τους με τους χρήστες. Παραδείγματα ψηφιακών βοηθών αποτελούν το BlueBot της αεροπορικής εταιρείας KLM και το Ask Mercedes της αυτοκινητοβιομηχανίας Mercedes-Benz.

- **ιατρική διάγνωση:** η τεχνητή νοημοσύνη χρησιμοποιείται στην ανάλυση δεδομένων από ασθενείς, όπως εικόνες από ακτινογραφίες και μαγνητικές τομογραφίες, για την αναγνώριση παθολογιών και την πρόβλεψη ασθενειών. Έτσι, η τεχνητή νοημοσύνη συμβάλλει στην πρόληψη, τη διάγνωση και τη θεραπεία ασθενειών, βελτιώνοντας την ποιότητα της υγειονομικής περίθαλψης.
- **σχεδιασμός διαδρομής** : υπηρεσίες όπως το Google Maps χρησιμοποιούν την τεχνητή νοημοσύνη για να αναλύουν τις συνθήκες κυκλοφορίας και να παρέχουν τις γρηγορότερες διαδρομές, βοηθώντας τους οδηγούς να εξικονομήσουν χρόνο και καύσιμα.
- **ανάπτυξη παιχνιδιών:** αλγόριθμοι τεχνητής νοημοσύνης χρησιμοποιούνται για τη δημιουργία χαρακτήρων που δεν ελέγχονται από τον παίκτη (*Non Player Characters*) με ευφυή συμπεριφορά, που περιλαμβάνει την προσαρμογή στις ενέργειες του παίκτη και την αναγνώριση των προτιμήσεών του, ώστε να προσφέρουν μια πιο ρεαλιστική και διασκεδαστική εμπειρία.

## 3.2 Μηχανική Μάθηση

### 3.2.1 Ορισμός

Ο όρος «Μηχανική Μάθηση» επινοήθηκε από τον Arthur Samuel το 1959 και περιγράφηκε ως «η ικανότητα ενός υπολογιστή να μαθαίνει χωρίς να προγραμματιστεί ρητά» (Samuel 1959). Ο Tom Mitchell έδωσε το 1997 έναν διάσημο, πιο μαθηματικό ορισμό των αλγορίθμων μηχανικής μάθησης: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από την εμπειρία E ως προς μια κλάση εργασιών T και μέτρο απόδοσης P, αν η απόδοσή του στις εργασίες της κλάσης T, όπως μετράται από το μέτρο απόδοσης P, βελτιώνεται με την εμπειρία E» (Mitchell 1997).

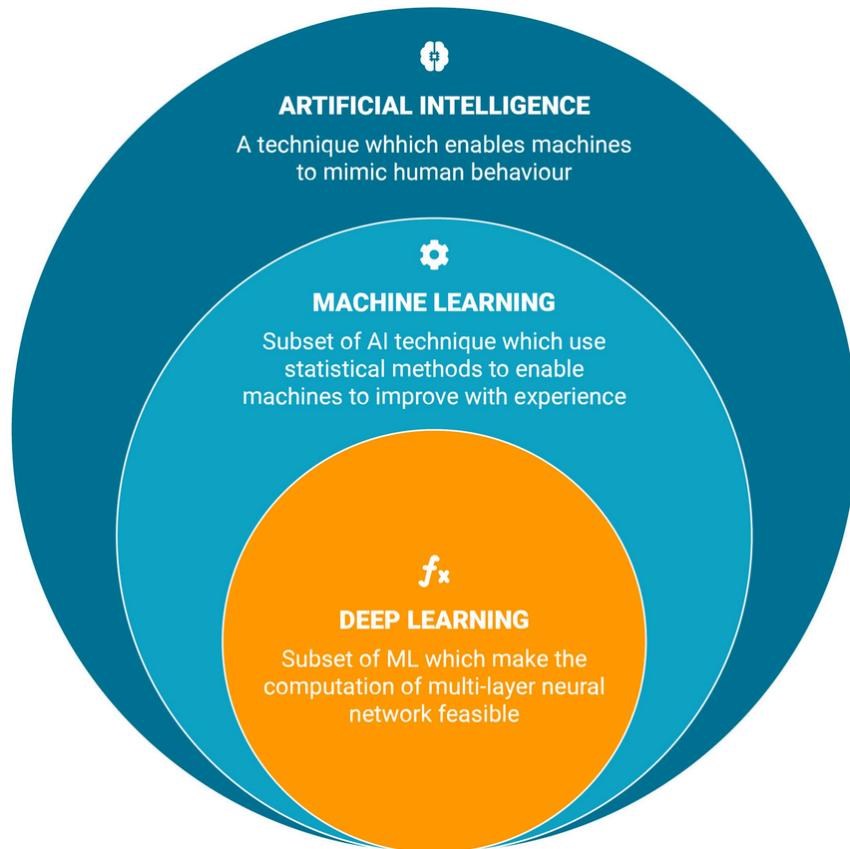
Επομένως, γίνεται σαφές ότι η μηχανική μάθηση ασχολείται με την ανάπτυξη αλγορίθμων, οι οποίοι μαθαίνουν να εκτελούν εργασίες μέσα από δεδομένα ή από προηγούμενη εμπειρία τους και όχι μέσω συγκεκριμένων εντολών. Αυτό είναι πολύ σημαντικό, καθώς τα παραγόμενα μοντέλα μηχανικής μάθησης είναι σε θέση να γενικεύουν τις γνώσεις τους και να τις εφαρμόζουν σε νέα δεδομένα. Το χαρακτηριστικό τους αυτό, τα κάνει ιδιαίτερα χρήσιμα για την επίλυση προβλημάτων, τα οποία δεν μπορούν να επιλυθούν με την κλασική προγραμματιστική προσέγγιση. Ωστόσο, η εκπαίδευση των μοντέλων μηχανικής μάθησης απαιτεί μεγάλο όγκο δεδομένων, καθώς και χρόνο και πόρους για την επεξεργασία τους, ενώ το τελικό αποτέλεσμα εξαρτάται σε μεγάλο βαθμό από την ποιότητα των δεδομένων. Συγκεκριμένα, τα μοντέλα μπορεί να αναπτύξουν προκαταλήψεις (*bias*), εφόσον αυτές υπάρχουν στα δεδομένα εκπαίδευσης τους. Για παράδειγμα, η μελέτη των (Thomson και Thomas

### 3.2 Μηχανική Μάθηση

2023), ανακάλυψε κοινωνικές προκαταλήψεις στην εφαρμογή παραγωγής εικόνων Midjourney. Όταν ζητήθηκε η παραγωγή εικόνων ανθρώπων σε εξειδικευμένα επαγγέλματα, τα αποτελέσματα απεικόνιζαν πάντα άνδρες, ενισχύοντας τη φύλετική προκατάληψη του ρόλου των γυναικών στον χώρο εργασίας.

#### 3.2.2 Κατάταξη πεδίου

Σήμερα, πολλοί συγχέουν τους όρους «Τεχνητή Νοημοσύνη» και «Μηχανική Μάθηση». Οφείλει να γίνει κατανοητό, πως η μηχανική μάθηση αποτελεί ένα υποσύνολο της τεχνητής νοημοσύνης, που εστιάζει στη βελτίωση της απόδοσης ενός συστήματος με βάση την εμπειρία. Ορισμένα συστήματα τεχνητής νοημοσύνης χρησιμοποιούν μεθόδους μηχανικής μάθησης, ενώ άλλα όχι. Αυτό φαίνεται με παραστατικό τρόπο, στην *Εικόνα 3.2*.



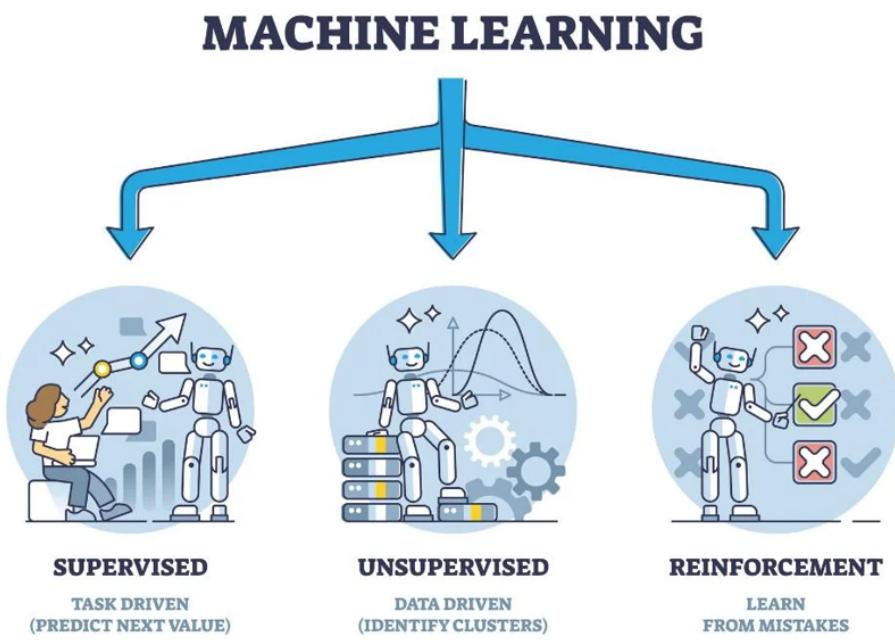
**Εικόνα 3.2.** Ιεραρχία πεδίων τεχνητής νοημοσύνης (Zand κ.ά. 2022).

Μάλιστα, η εικόνα 3.2 παρουσιάζει και τη σχέση μεταξύ μηχανικής μάθησης και βαθιάς μάθησης, δείχνοντας πως η βαθιά μάθηση αποτελεί μια περαιτέρω εξειδικευση της μηχανικής μάθησης. Αμφότερες οι τεχνικές αυτές, χρησιμοποιούν τεχνητά νευρωνικά δίκτυα για να «μάθουν» από τα

δεδομένα. Όμως, η βαθιά μάθηση χρησιμοποιεί πιο πολύπλοκα, πολυεπίπεδα νευρωνικά δίκτυα, τα οποία απαιτούν μεγαλύτερο όγκο δεδομένων και πόρων για την εκπαίδευσή τους.

### 3.2.3 Κατηγορίες

Η μηχανική μάθηση χωρίζεται σε τρεις κύριες κατηγορίες, ανάλογα με τον τρόπο εκπαίδευσης των μοντέλων: την επιβλεπόμενη μάθηση, την μη επιβλεπόμενη μάθηση και την ενισχυτική μάθηση. Οι τρεις αυτές κατηγορίες παρουσιάζονται στην *Εικόνα 3.3* και περιγράφονται παρακάτω.



Εικόνα 3.3. Κατηγορίες μηχανικής μάθησης (Stewart 2023).

#### Επιβλεπόμενη Μάθηση

Η Επιβλεπόμενη Μάθηση (*Supervised Learning*) είναι ο πιο συνηθισμένος τύπος μηχανικής μάθησης. Πήρε το όνομα της, καθώς η εκπαίδευση γίνεται υπό επίβλεψη, δηλαδή το μοντέλο μαθαίνει μέσω παραδειγμάτων. Συγκεκριμένα, παρέχεται στο μοντέλο ένα σύνολο δεδομένων εκπαίδευσης με ετικέτες (*labeled data*). Αυτό σημαίνει ότι κάθε δεδομένο, αποτελείται από ένα ζεύγος εισόδου-επιθυμητής εξόδου. Το μοντέλο κατά την εκπαίδευση, εντοπίζει μοτίβα στα δεδομένα και προβλέπει για κάθε είσοδο, ποιά είναι η αντίστοιχη έξοδος. Όταν κάνει λάθος, το μοντέλο αναπροσαρμόζεται, μέχρι να μάθει να αντιστοιχίζει σωστά τις εισόδους στις αντίστοιχες έξοδους. Το ζητούμενο είναι, το τελικό μοντέλο να μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις σε νέα δεδομένα -δηλαδή

### 3.2 Μηχανική Μάθηση

---

δεδομένα που δεν υπήρχαν στο σύνολο εκπαίδευσης- και να πετυχαίνει σε αυτά μεγάλο ποσοστό επιτυχίας.

Κλασικό παράδειγμα επιβλεπόμενης μάθησης αποτελεί η αναγνώριση του φύλου από εικόνες. Στο σύνολο δεδομένων εκπαίδευσης, κάθε εικόνα έχει ετικέτα με το φύλο του ατόμου που απεικονίζεται. Το μοντέλο εκπαιδεύεται να αναγνωρίζει τα χαρακτηριστικά που διαφοροποιούν τα δύο φύλα και να κάνει προβλέψεις για το φύλο του ατόμου. Υπάρχει περίπτωση, το μοντέλο να κάνει σωστές προβλέψεις στις εικόνες που εκπαιδεύτηκε, αλλά όχι σε άγνωστες εικόνες. Τότε, το μοντέλο δεν έχει μάθει να αναγνωρίζει σωστά το φύλο, αλλά απλά έχει μάθει να απαντάει σωστά στα δεδομένα εκπαίδευσης του. Το φαινόμενο αυτό ονομάζεται υπερπροσαρμογή ή υπερεκπαίδευση (*overfitting*) του μοντέλου και προφανώς, αποτελεί ανεπιθύμητη συμπεριφορά.

Επίσης, αξίζει να σημειωθεί πως όταν η έξοδος είναι μία τιμή από ένα πεπερασμένο σύνολο τιμών (όπως π.χ. πριν, άντρας/γυναίκα), τότε το πρόβλημα μάθησης ονομάζεται Ταξινόμηση (*Classification*). Αντίθετα, όταν η έξοδος είναι μία συνεχής τιμή (π.χ. η αυριανή θερμοκρασία), τότε το πρόβλημα ονομάζεται Παλινδρόμηση (*Regression*). Συνήθεις αλγόριθμοι που χρησιμοποιούνται για προβλήματα ταξινόμησης είναι τα Δέντρα Απόφασης (*Decision Trees*) και οι Μηχανές Διανυσμάτων Υποστήριξης (*Support Vector Machines*). Αντίθετα, για προβλήματα παλινδρόμησης χρησιμοποιούνται αλγόριθμοι όπως η Γραμμική Παλινδρόμηση (*Linear Regression*) και η Πολυωνυμική Παλινδρόμηση (*Polynomial Regression*).

### Μη Επιβλεπόμενη Μάθηση

Η μη επιβλεπόμενη μάθηση (*Unsupervised Learning*) αναφέρεται στις περιπτώσεις όπου τα δεδομένα δεν έχουν ετικέτες, δηλαδή το μοντέλο δεν γνωρίζει την επιθυμητή έξοδο για κάθε δεδομένο. Πλέον, στόχος είναι η αναγνώριση μοτίβο, δομών ή συσχετίσεων στα δεδομένα. Οι πιο συνηθισμένες εργασίες μη επιβλεπόμενης μάθησης είναι η ομαδοποίηση και η μείωση διαστάσεων.

Η Ομαδοποίηση (*Clustering*) αφορά την οργάνωση των δεδομένων, χωρίζοντας τα σε ομάδες με παρόμοια χαρακτηριστικά. Ένα παράδειγμα αποτελεί η οργάνωση των πελατών μίας επιχείρησης σε ομάδες με βάση το ιστορικό αγορών τους, έτσι ώστε να εφαρμοστεί διαφορετική στρατηγική προώθησης για κάθε ομάδα. Παραδείγματα αλγορίθμων ομαδοποίησης είναι ο K-Means και η Ιεραρχική Ομαδοποίηση (*Hierarchical Clustering*).

Η Μείωση Διαστάσεων (*Dimensionality Reduction*) αφορά τη μείωση του αριθμού των χαρακτηριστικών που περιγράφουν τα δεδομένα, διατηρώντας όμως σε μεγάλο βαθμό την πληροφορία που περιέχουν. Αυτό είναι χρήσιμο για την απλοποίηση μοντέλων, την αφαίρεση θορύβου από τα δεδομένα και την οπτικοποίηση τους. Ένας από τους πιο διάσημους αλγορίθμους μείωσης διαστάσεων είναι ο PCA (*Principal Component Analysis*).

## Ενισχυτική Μάθηση

Η ενισχυτική μάθηση (*Reinforcement Learning*) αποτελεί την κατηγορία της μηχανικής μάθησης που μιμείται πιο πιστά τον τρόπο μάθησης των ανθρώπων. Ένας πράκτορας ενισχυτικής μάθησης εκπαιδεύεται μέσω της αλληλεπίδρασης με το περιβάλλον του και επιλέγοντας δράσεις σε αυτό. Ο πράκτορας λαμβάνει θετική ανταμοιβή, όταν οι ενέργειες του οδηγούν σε επιθυμητά αποτελέσματα και αρνητική ανταμοιβή, όταν οδηγούν σε ανεπιθύμητα αποτελέσματα. Μέσω δοκιμών και λαθών, ο πράκτορας μαθαίνει να παίρνει αποφάσεις που μεγιστοποιούν τις ανταμοιβές του. Η ενισχυτική μάθηση χρησιμοποιείται σε εφαρμογές όπως η ρομποτική, τα παιχνίδια και η διαχείριση πόρων. Μερικοί δημοφιλείς αλγόριθμοι ενισχυτικής μάθησης είναι ο Q-Learning, ο PPO (*Proximal Policy Optimization*) και ο SAC (*Soft Actor-Critic*).

Η ενισχυτική μάθηση αποτελεί την κατηγορία των αλγορίθμων μηχανικής μάθησης που χρησιμοποιούνται στην παρούσα εργασία και για αυτό, αναλύεται σε μεγαλύτερο βάθος στην επόμενη ενότητα.

## 3.3 Ενισχυτική Μάθηση

### 3.3.1 Γενική επισκόπηση

#### Κεντρική ιδέα

Η Ενισχυτική Μάθηση (*Reinforcement Learning*) αποτελεί την κατηγορία της μηχανικής μάθησης, στην οποία ένας πράκτορας μαθαίνει από την εμπειρία του, καθώς αλληλεπιδράει με το περιβάλλον του. Ο πράκτορας ενισχυτικής μάθησης επιλέγει ελεύθερα ενέργειες και δέχεται ανάδραση για αυτές, υπό μορφή ανταμοιβής. Συγκεκριμένα, ο πράκτορας λαμβάνει θετική ανταμοιβή (επιβράβευση), όταν οι ενέργειες του οδηγούν σε επιθυμητά αποτελέσματα και αρνητική ανταμοιβή (τιμωρία), όταν οδηγούν σε ανεπιθύμητα αποτελέσματα. Έτσι, μέσω δοκιμών και λαθών (*trial and error*), ο πράκτορας μαθαίνει σταδιακά να παίρνει αποφάσεις που μεγιστοποιούν τις ανταμοιβές του. Η διαδικασία αυτή της εκπαίδευσης διαφέρει σημαντικά σε σχέση με τις δύο προηγούμενες κατηγορίες μηχανικής μάθησης, αφού πλέον ο αλγόριθμος εκπαιδεύεται, χωρίς κάποιο σταθερό σύνολο δεδομένων εισόδου. Αντίθετα, η εκπαίδευση στην ενισχυτική μάθηση είναι μία δυναμική διαδικασία, στην οποία ο πράκτορας εκτελεί ενέργειες και μεταβάλλει το περιβάλλον του. Επομένως, πρόκειται για μία ενεργή διαδικασία μάθησης, η οποία θυμίζει τον τρόπο με τον οποίο οι ζωντανοί οργανισμοί μαθαίνουν.

Πράγματι, ας αναλογιστούμε το παράδειγμα της εκπαίδευσης ενός σκύλου. Στο σενάριο μας, ο ιδιοκτήτης του σκύλου επιθυμεί να του μάθει την ενέργεια «κάτσε». Έτσι, ο ιδιοκτήτης του σκύλου δείχνει με το χέρι του το έδαφος και φωνάζει προς τον σκύλο «κάτσε». Όσο ο σκύλος στέκεται όρθιος, ο ιδιοκτήτης του δίνει αρνητική ανταμοιβή (π.χ. του φωνάζει «κακό σκυλί», έχοντας τεντωμένο

### 3.3 Ενισχυτική Μάθηση

---

τον δείκτη του και δείχνοντας προς αυτό). Όταν ο σκύλος κάθεται, ο ιδιοκτήτης του δίνει θετική ανταμοιβή (π.χ. του χαϊδεύει το κεφάλι ή του δίνει μία λιχουδιά). Έτσι, ο σκύλος μαθαίνει σταδιακά, πως όταν παρατηρεί τον άνθρωπο στην κατάσταση «κάτσε» (δηλαδή χέρι προς το έδαφος και προφορική εντολή), η ενέργεια του να καθίσει στο έδαφος του προσφέρει θετική ανταμοιβή και για αυτό την επιλέγει.

Γίνεται πλέον σαφές, πως στόχος της ενισχυτικής μάθησης είναι η εκπαίδευση του πράκτορα, ώστε να προβαίνει σε ενέργειες που μεγιστοποιούν την ανταμοιβή του. Μάλιστα, μία σημαντική παρατήρηση είναι πως ο πράκτορας πρέπει να μάθει να μεγιστοποιεί τη συνολική ανταμοιβή του, δηλαδή να σκέπτεται μακροπρόθεσμα. Συγκεκριμένα, ενδέχεται μία ενέργεια του πράκτορα να οδηγήσει σε αρνητική ανταμοιβή άμεσα, όμως σε βάθος χρόνου να βοηθήσει τον πράκτορα να πετύχει μεγάλη θετική ανταμοιβή. Για παράδειγμα, στο παιχνίδι του σκακιού υπάρχει η στρατηγική της θυσίας (*sacrifice*), στην οποία ο παίκτης επιλέγει να χάσει ένα κομμάτι (π.χ. την βασίλισσα του), για να πετύχει στο μέλλον κάτι μεγαλύτερης αξίας (π.χ. ρουά-ματ στον αντίπαλο βασιλιά).

### Πλεονεκτήματα

Η εκπαίδευση με ενισχυτική μάθηση έχει ορισμένα σημαντικά πλεονεκτήματα σε σχέση με τις πιο παραδοσιακές μεθόδους μηχανικής μάθησης, τα οποία περιγράφονται παρακάτω:

- Δεν χρειάζεται μεγάλα σύνολα δεδομένων: σε πολλά πεδία είναι δύσκολο να συλλεχθούν δεδομένα ή δεν υπάρχουν στον βαθμό που απαιτείται για μία αποδοτική εκπαίδευση επιβλεπόμενης μάθησης. Η ενισχυτική μάθηση αποτελεί μία εναλλακτική λύση, καθώς ο πράκτορας μαθαίνει μόνος του, χωρίς προηγούμενα δεδομένα.
- Δεν απαιτεί γνώση ειδικού στο πεδίο εφαρμογής: στην επιβλεπόμενη μάθηση, τα δεδομένα αποτελούν ζεύγη εισόδου-επιθυμητής εξόδου. Για παράδειγμα, για την εκπαίδευση ενός μοντέλου να παίζει σκάκι, τα δεδομένα θα ήταν της μορφής: κατάσταση σκακιέρας-κίνηση που έκανε ο παίκτης. Έτσι, το μοντέλο θα μάθαινε να παίζει σκάκι με τον τρόπο που παίζουν οι παίκτες στα δεδομένα εκπαίδευσης. Άρα, θα έπρεπε αυτοί οι παίκτες να είναι ειδικοί στο παιχνίδι, προκειμένου να πετύχει το μοντέλο υψηλή απόδοση. Αντίθετα, στην ενισχυτική μάθηση, οι σχεδιαστές ενός συστήματος αρκεί να έχουν βασική γνώση του πεδίου εφαρμογής του, για να εκπαιδεύσουν επιτυχώς έναν πράκτορα. Για παράδειγμα, στην περίπτωση του σκακιού, αρκεί οι σχεδιαστές να γνωρίζουν τους κανόνες του παιχνιδιού, ώστε π.χ. να επιβραβεύουν τον πράκτορα όταν κερδίζει κομμάτια του αντιπάλου και να τον τιμωρούν όταν χάνει κομμάτια του.
- Προσφέρει καινοτόμες λύσεις: η επιβλεπόμενη μάθηση ουσιαστικά μιμείται τα δεδομένα εκπαίδευσης. Ναι μεν μπορεί να πετύχει υψηλότερες επιδόσεις από τον άνθρωπο (π.χ. σκάκι), όμως δεν μπορεί να μάθει μία εντελώς νέα προσέγγιση για την επίλυση του προβλήματος. Αντίθετα, οι αλγόριθμοι ενισχυτικής μάθησης μπορούν να προτείνουν εντελώς νέες και

επαναστατικές λύσεις, τις οποίες δεν είχε σκεφτεί ποτέ κάποιος άνθρωπος. Ένα τέτοιο αξιοσημείωτο παράδειγμα συνέβη στη νική του AlphaGo, ενός συστήματος ενισχυτικής μάθησης, έναντι του παγκόσμιου πρωταθλητή στο Go, Lee Sedol. Κατά τη διάρκεια του αγώνα, το AlphaGo έκανε μία ασυνήθιστη κίνηση (κίνηση 37) που αρχικά θεωρήθηκε λάθος από τους ειδικούς στο παιχνίδι. Μάλιστα, υπολογίστηκε ότι η πιθανότητα ένας άνθρωπος να προέβαινε σε αυτή την κίνηση, στη συγκεκριμένη θέση, είναι ίση με 0.0001%. Ωστόσο, η κίνηση αποδείχθηκε εν τέλει δημιουργική και καθοριστική για τη νίκη του AlphaGo (Metz 2016).

- Είναι κατάλληλη για περιβάλλοντα **σειριακής λήψης αποφάσεων**: η ενισχυτική μάθηση μαθαίνει στον πράκτορα να μεγιστοποιεί τη συνολική ανταμοιβή σε βάθος χρόνου. Έτσι, είναι κατάλληλη για σενάρια, όπου οι αποφάσεις είναι σειριακές και το αποτέλεσμα μίας ενέργειας επηρεάζει τις μελλοντικές αποφάσεις. Αντίθετα, στην επιβλεπόμενη μάθηση, το μοντέλο πραγματοποιεί ανεξάρτητες προβλέψεις σε κάθε βήμα, χωρίς να λαμβάνει υπόψη το πως αυτές θα επηρεάσουν τις μελλοντικές αποφάσεις.

Ωστόσο, η ενισχυτική μάθηση έχει και μειονεκτήματα, τα οποία δυσκολεύουν την επιτυχή εφαρμογή της. Πολλά από αυτά τα προβλήματα προέκυψαν και κατά τη διάρκεια της εκπαίδευσης του πράκτορα αυτόματης στάθμευσης. Για αυτό, περιγράφονται στο κεφάλαιο της εκπαίδευσης, στην *Ενότητα 5.5*.

## Εφαρμογές

Η ενισχυτική μάθηση αποτελεί ουσιαστικά, την επιστημή της λήψης αποφάσεων κι ως εκ τούτου, έχει εφαρμογές σε πολλά πεδία. Στο πεδίο των χρηματοοικονομικών, η εταιρία Goldman Sachs έχει ξεκινήσει τη χρήση της ενισχυτικής μάθησης στις πλατφόρμες συναλλαγών της, για βελτιώσει την απόδοση των επενδυτικών στρατηγικών της (McMurray 2023). Στον τομέα της βιοτεχνολογίας, η εταιρία Atomwise χρησιμοποιεί την ενισχυτική μάθηση στην πλατφόρμα της AtomNet, που χρησιμοποιείται για την ανακάλυψη νέων φαρμάκων (Atomwise 2018). Στον χώρο της ρομποτικής, η εταιρία Boston Dynamics έχει ενσωματώσει την ενισχυτική μάθηση στα συστήματα ελέγχου του τετράποδου ρομπότ της, Spot και έχει βελτιώσει την αυτόνομη πλοιόγηση του σε πολύπλοκα περιβάλλοντα (BostonDynamics 2024).

Ωστόσο, το επικρατέστερο πεδίο εφαρμογής των αλγορίθμων ενισχυτικής μάθησης είναι τα παιχνίδια. Αυτό οφείλεται στο γεγονός ότι τα παιχνίδια αποτελούν έναν ασφαλές χώρο εκπαίδευσης πρακτόρων και ανάπτυξης αλγορίθμων, προτού αυτοί εφαρμοστούν σε προβλήματα του πραγματικού κόσμου. Μάλιστα, σε περιβάλλοντα προσομοιώσεων δεν υπάρχουν χρονικοί περιορισμοί, οπότε ο πράκτορας μπορεί να εκπαιδεύεται ασταμάτητα για μεγάλα χρονικά διαστήματα, κάτι που συχνά είναι απαραίτητο για την επιτυχία της ενισχυτικής μάθησης. Επίσης, τα παιχνίδια απαιτούν σε σημαντικό βαθμό νοητικές ικανότητες από τον παίκτη, κι έτσι αποτελούν μία καλή πλατφόρμα εκπαίδευσης αλγορίθμων τεχνητής νοημοσύνης.

### 3.3 Ενισχυτική Μάθηση

---

Οι πρώτες απόπειρες έγιναν το 1959 από τον Arthur Samuel, ο οποίος ανέπτυξε ένα πρόγραμμα που έπαιζε το παιχνίδι της ντάμας (Samuel 1959). Στη συνέχεια, το 1992, ο Gerald Tesauro ανέπτυξε τον αλγόριθμο TD-Gammon με τη βοήθεια νευρωνικών δικτύων, ο οποίος έπαιζε τάβλι και κατάφερε να φτάσει σε επίπεδο ανάλογο των τριών κορυφαίων παικτών στον κόσμο (Tesauro 1994).

Ωστόσο, μέχρι και τη δεκαετία του 2010, υπήρχαν σημαντικοί περιορισμοί στην εφαρμογή της ενισχυτικής μάθησης σε πολύπλοκα προβλήματα ή προβλήματα μεγάλης διαστασιμότητας. Παρόλο που είχαν ήδη αναπτυχθεί αλγόριθμοι επίλυσης τέτοιων προβλημάτων, η μικρή διαθέσιμη υπολογιστική ισχύς της εποχής δεν επέτρεπε την εκπαίδευση των μοντέλων σε λογικά χρονικά διαστήματα. Το πρόβλημα αυτό υπερκεράστηκε από την τεχνολογική ανάπτυξη στον τομέα του υλικού (*hardware*) με χαρακτηριστικό παράδειγμα την παραλληλοποίηση των υπολογισμών στις μονάδες επεξεργασίας γραφικών (*GPU*), οι οποίες επιταχύνουν σημαντικά τη διαδικασία της εκπαίδευσης. Έτσι, ξεκίνησε η εποχή της βαθιάς ενισχυτικής μάθησης και άρχισαν να φαίνονται για πρώτη φορά οι πραγματικές δυνατότητες των τεχνητών νευρωνικών δικτύων. Ορισμένες επιτυχίες-ορόσημα της βαθιάς ενισχυτικής μάθησης σε περιβάλλοντα παιχνιδιών αναφέρονται με χρονολογική σειρά παρακάτω:

- 2012: η εταιρία DeepMind της Google ανέπτυξε το πρώτο σύγχρονο σύστημα βαθιάς ενισχυτικής μάθησης, τον αλγόριθμο DQN (*Deep Q-Network*). Ο αλγόριθμος αυτός, εκπαιδεύτηκε ξεχωριστά στα 49 παιχνίδια της πλατφόρμας Atari2600, δεχόμενος ως είσοδο μόνο τα pixels της οθόνης και το σκορ του παιχνιδιού και κατάφερε να φτάσει σε επίπεδο συγκρίσιμο με αυτό ενός επαγγελματία δοκιμαστή παιχνιδιών (Mnih κ.ά. 2015).
- 2016: η εταιρία DeepMind παρουσίασε τον αλγόριθμο AlphaGo, για το παιχνίδι στρατηγικής Go (DeepMind 2016). Το Go, είναι ένα παιχνίδι πολύ πολύπλοκο από το σκάκι, έχοντας σημαντικά μεγαλύτερο χώρο καταστάσεων κι έτσι, αποτελούσε πρόκληση για την τεχνητή νοημοσύνη. Οι παραδοσιακοί αλγόριθμοι μηχανικής μάθησης δυσκολεύονταν να αξιολογήσουν όλες τις πιθανές κινήσεις, να αναπτύξουν ανθρώπινη δημιουργικότητα και συνολικά, να ανταγωνιστούν τους ανθρώπους. Όμως, ο αλγόριθμος AlphaGo κατάφερε να νικήσει σε μία σειρά παιχνιδιών τον θρυλικό παγκόσμιο πρωταθλητή στο Go, Lee Sedol. Αυτή η νίκη αποτέλεσε απόδειξη πως τα συστήματα βαθιάς ενισχυτικής μάθησης μπορούν να μάθουν να λύνουν τα πιο δύσκολα προβλήματα, σε υψηλά περίπλοκα περιβάλλοντα.
- 2017: η εταιρία OpenAI ανέπτυξε μία παραλλαγή του αλγορίθμου Proximal Policy Optimization, τον αλγόριθμο OpenAI Five για την εκπαίδευση πρακτόρων στο παιχνίδι Dota2, ένα πολυπρακτορικό παιχνίδι (παίζεται από 2 ομάδες των 5 ατόμων), αβέβαιης πληροφορίας και με ιδιαίτερα πολύπλοκες καταστάσεις και ενέργειες. Οι πράκτορες του OpenAI Five εκπαιδεύτηκαν για 10 μήνες και μέσω self-play, δηλαδή παίζοντας μεταξύ τους, και το 2019 κατάφεραν να νικήσουν τους τρέχοντες παγκόσμιους πρωταθλητές στο παιχνίδι (OpenAI 2019).
- 2020: η εταιρία DeepMind παρουσίασε τον Agent57, μία βελτιωμένη έκδοση του αλγορίθμου

DQN, ο οποίος χρησιμοποιεί έναν μετα-ελεγκτή για την προσαρμογή της εξερεύνησης και τη ρύθμιση της μακροπρόθεσμης έναντι της βραχυπρόθεσμης συμπεριφοράς του πράκτορα (Puigdomenech κ.ά. 2020). Ο Agent57 εκπαιδεύτηκε στα 57 παιχνίδια της πλατφόρμας Atari2600 και κατάφερε να ξεπεράσει τις επιδόσεις επαγγελματιών παικτών σε κάθε ένα από αυτά. Το σημαντικό σε αυτή την επιτυχία, είναι πως ήταν η πρώτη φορά που ένας αλγόριθμος κατάφερε κάτι ανάλογο στο σύνολο των 57 παιχνιδιών, τα οποία διακρίνονται για την πολυπλοκότητα και τη διαφορετικότητά τους.

### 3.3.2 Βασικές Έννοιες και Ορολογία

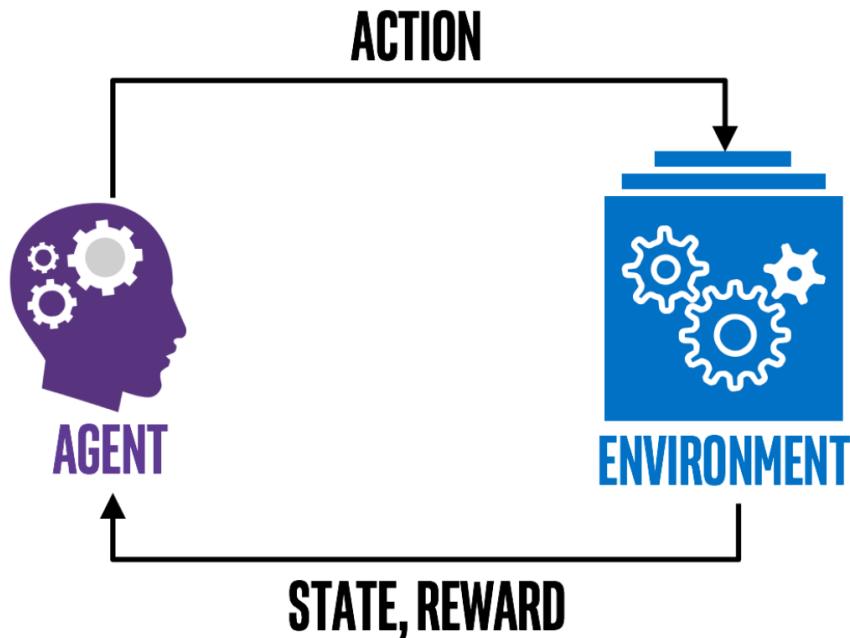
Έχοντας ήδη περιγράψει την γενικότερη ιδέα της ενισχυτικής μάθησης, στην υποενότητα αυτή θα μελετήσουμε το πεδίο σε μεγαλύτερο βάθος, παρουσιάζοντας τις βασικές έννοιες και την ορολογία που χρησιμοποιείται. Αυτό θα βοηθήσει στην καλύτερη κατανόηση των βασικών αρχών της εκπαίδευσης πρακτόρων, τη μεθοδολογία που ακολουθείται, καθώς και τους στόχους, αλλά και τα προβλήματα που προκύπτουν κατά την εκπαίδευση.

#### Κύκλος Ενισχυτικής Μάθησης

Ο κύκλος της ενισχυτικής μάθησης περιλαμβάνει την εξής διαδικασία: ο πράκτορας επιλέγει μία ενέργεια και την εκτελεί. Έτσι, μεταβάλει το περιβάλλον, δηλαδή αυτό μεταβαίνει σε μία νέα κατάσταση. Έπειτα, ο πράκτορας δέχεται ως είσοδο τη νέα κατάσταση του περιβάλλοντος, καθώς και την ανταμοιβή που προέκυψε από την ενέργειά του. Με βάση αυτές τις πληροφορίες, ο πράκτορας επιλέγει την επόμενη ενέργεια που θα εκτελέσει. Ο κύκλος αυτός φαίνεται και παραστατικά στην *Εικόνα 3.4*.

Μία επανάληψη της παραπάνω διαδικασίας ονομάζεται και βήμα (*step*) της εκπαίδευσης. Συνηθίζεται η εκπαίδευση ενός πράκτορα να χαρακτηρίζεται από το πλήθος των βημάτων στα οποία ο πράκτορας έχει εκπαιδευτεί. Στην πράξη, χρειάζονται μερικά εκατομμύρια βήματα εκπαίδευσης προκειμένου ο πράκτορας να φτάσει σε επιθυμητή απόδοση.

Επομένως, γίνεται πλέον κατανοητό πως κάθε πρόβλημα ενισχυτικής μάθησης αποτελείται από δύο βασικές οντότητες: το **περιβάλλον** και τον **πράκτορα**, καθώς επίσης και τρία κανάλια επικοινωνίας αυτών: της **ανταμοιβής**, των **καταστάσεων** και των **ενεργειών**. Ο όρος **πράκτορας** έχει ήδη αναλυθεί στην υποενότητα *3.1.2*, ενώ οι υπόλοιποι όροι περιγράφονται λεπτομερώς στις επόμενες παραγγράφους.



**Εικόνα 3.4.** Κύκλος Ενισχυτικής Μάθησης (Lee 2019).

### Περιβάλλον

Το περιβάλλον αποτελεί τον κόσμο, στον οποίο ο πράκτορας εκτελεί ενέργειες. Ένα περιβάλλον ενισχυτικής μάθησης πρέπει να ικανοποιεί την Μαρκοβιανή ιδιότητα. Προκειμένου να γίνει κατανοητή αυτή η ιδιότητα, πρέπει πρώτα να γίνει αναφορά στις Διαδικασίες Αποφάσεων Μαρκόβ.

### Διαδικασίες Αποφάσεων Μαρκόβ

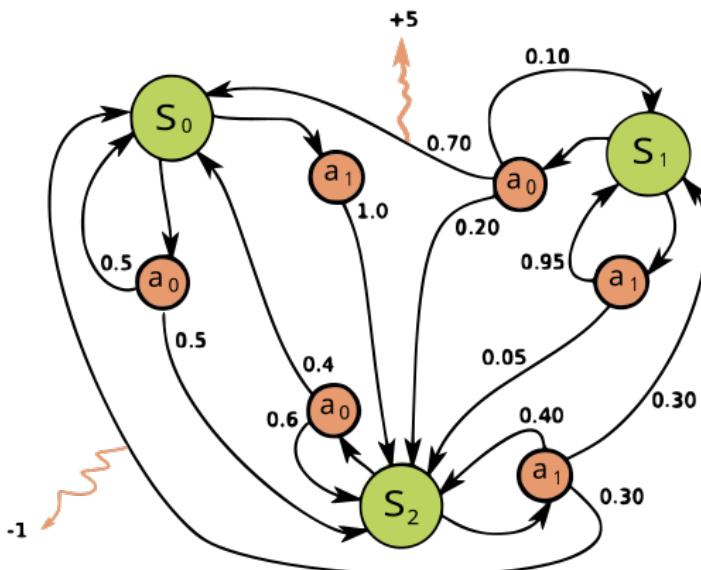
Οι διαδικασίες αποφάσεων Μαρκόβ (Markov Decision Processes - MDPs) αποτελούν ένα μαθηματικό πλαίσιο που χρησιμοποιείται για την περιγραφή ενός περιβάλλοντος σε προβλήματα επίτευξης ενός στόχου. Παρέχει μία μοντελοποίηση της λήψης αποφάσεων, σε καταστάσεις όπου τα αποτελέσματα είναι μερικώς τυχαία και μερικώς υπό τον έλεγχο του λήπτη αποφάσεων (δηλαδή του πράκτορα). Αποτελούν επέκταση των αλυσίδων Μαρκόβ, με τη διαφορά ότι προσθέτουν ενέργειες και ανταμοιβές. Έτσι, τα MDPs χρησιμοποιούνται για να περιγράψουν ένα περιβάλλον στο οποίο θέλουμε να εφαρμόσουμε ενισχυτική μάθηση.

Περιγράφονται από την εξής διαδικασία: Ο πράκτορας αλληλεπιδρά με το περιβάλλον σε μία ακολουθία χρονικών στιγμών  $t = 0, 1, 2, \dots$ . Σε κάθε χρονική στιγμή  $t$ , το περιβάλλον βρίσκεται σε μία κατάσταση  $s_t \in S$ , όπου  $S$  είναι το σύνολο των καταστάσεων του περιβάλλοντος. Η κατάσταση αυτή δίνεται ως είσοδος στον πράκτορα, ο οποίος στη συνέχεια επιλέγει μία ενέργεια  $a_t \in A$ , όπου  $A$  είναι το σύνολο των ενεργειών που μπορεί να εκτελέσει. Το περιβάλλον ανταποκρίνεται στην ενέργεια του πράκτορα, μεταβαίνοντας σε μία νέα κατάσταση  $s_{t+1}$  και επιστρέφοντας στον

πράκτορα μία ανταμοιβή  $r_{t+1} \in R$ , όπου  $R$  είναι το σύνολο των πιθανών ανταμοιβών. Με τον τρόπο αυτό, μία διαδικασία απόφασης Μαρκόβ ορίζεται ως μία λίστα 4 στοιχείων  $(S, A, P, R)$ , όπου:

- $S$  είναι το σύνολο των καταστάσεων του περιβάλλοντος (καλείται και χώρος καταστάσεων),
- $A$  είναι το σύνολο των ενεργειών που μπορεί να εκτελέσει ο πράκτορας (καλείται και χώρος ενεργειών),
- $P$  είναι η συνάρτηση πιθανότητας μετάβασης, όπου  $P(s_{t+1}|s_t, a_t)$  είναι η πιθανότητα να μεταβεί το περιβάλλον στην κατάσταση  $s_{t+1}$  μετά την εκτέλεση της ενέργειας  $a_t$  στην κατάσταση  $s_t$ ,
- $R$  είναι η συνάρτηση ανταμοιβής, ενώ το  $R_{a_t}(s_t, s_{t+1})$  είναι η ανταμοιβή που λαμβάνει ο πράκτορας όταν μεταβεί από την κατάσταση  $s_t$  στην κατάσταση  $s_{t+1}$  μετά την εκτέλεση της ενέργειας  $a_t$ .

Ένα παράδειγμα μίας Διαδικασίας Απόφασης Μαρκόβ φαίνεται με τη χρήση ενός γράφου στην *Εικόνα 3.5*. Οι πράσινοι κόμβοι αντιστοιχούν στις καταστάσεις, οι πορτοκαλί κόμβοι στις ενέργειες, και τα βάρη των ακμών στις πιθανότητες μετάβασης. Επίσης, τα πορτοκαλί βέλη αντιστοιχούν στις ανταμοιβές που λαμβάνει ο πράκτορας μετά την εκτέλεση της ενέργειας.



Εικόνα 3.5. Παράδειγμα Διαδικασίας Απόφασης Μαρκόβ (Alvarez 2017)

### Μαρκοβιανή Ιδιότητα

Η μαρκοβιανή ιδιότητα αναφέρεται στην ιδιότητα «αμνησίας» μίας στοχαστικής διαδικασίας, δηλαδή στο χαρακτηριστικό ότι η μελλοντική εξέλιξή της είναι ανεξάρτητη από το παρελθόν της.

### 3.3 Ενισχυτική Μάθηση

---

Συγκεκριμένα, δηλώνει ότι η επόμενη κατάσταση  $s_{t+1}$  εξαρτάται μόνο από την τρέχουσα κατάσταση  $s_t$  και την ενέργεια  $a_t$  που επιλέγει ο πράκτορας. Διοθέντος αυτών των δύο, είναι ανεξάρτητη από όλες τις προηγούμενες καταστάσεις και ενέργειες. Αυτό περιγράφεται μαθηματικά από την εξίσωση 3.1:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0) = P(s_{t+1}|s_t, a_t) \quad (3.1)$$

Η μαρκοβιανή ιδιότητα είναι κρίσιμη για την επιτυχή εφαρμογή της ενισχυτικής μάθησης, καθώς εξασφαλίζει ότι ο πράκτορας μπορεί να λάβει αποφάσεις με βάση μόνο την τρέχουσα κατάσταση του περιβάλλοντος, χωρίς να χρειάζεται να αποθηκεύσει όλες τις προηγούμενες καταστάσεις.

Πέραν όμως της μαρκοβιανής ιδιότητας, ένα περιβάλλον ενισχυτικής μάθησης περιγράφεται κι από άλλες ιδιότητες, οι οποίες οφείλονται στη διαφορετική φύση του κάθε προβλήματος και επηρεάζουν τη σημαντικά τη σχεδίαση του πράκτορα. Οι ιδιότητες αυτές είναι οι εξής:

#### Πλήρως ή μερικώς παρατηρήσιμο

Το περιβάλλον είναι πλήρως παρατηρήσιμο όταν ο πράκτορας έχει πρόσβαση στην πλήρη κατάστασή του κάθε χρονική στιγμή. Αντίθετα, όταν ο πράκτορας έχει πρόσβαση μόνο σε μέρος της κατάστασης του περιβάλλοντος, τότε το περιβάλλον είναι μερικώς παρατηρήσιμο. Ένα παράδειγμα ενός πλήρους παρατηρήσιμου περιβάλλοντος είναι το παιχνίδι του σκακιού, όπου οι παίκτες γνωρίζουν τη θέση όλων των πιονιών στο ταμπλό. Αντίθετα, ένα παράδειγμα μερικώς παρατηρήσιμου περιβάλλοντος είναι το παιχνίδι του πόκερ, όπου οι παίκτες δεν γνωρίζουν τις κάρτες των αντιπάλων τους.

#### Αιτιοκρατικό ή στοχαστικό

Αιτιοκρατικό χαρακτηρίζεται το περιβάλλον στο οποίο η επόμενη κατάστασή του προσδιορίζεται με ακρίβεια από την τρέχουσα κατάστασή του και την ενέργεια του πράκτορα. Στην περίπτωση αιτιοκρατικού περιβάλλοντος, όταν ο πράκτορας βρίσκεται σε συγκεκριμένη κατάσταση και εκτελεί μία συγκεκριμένη ενέργεια, θα προκύπτει πάντα η ίδια ανταμοιβή και η ίδια επόμενη κατάσταση. Για παράδειγμα, παιχνίδια όπως το σκάκι είναι αιτιοκρατικά, καθώς η κίνηση ενός πιονιού σε συγκεκριμένη θέση θα οδηγήσει πάντα σε μία συγκεκριμένη κατάσταση του παιχνιδιού.

Όταν δεν υπάρχει αυτή η νομοτελειακή σχέση μεταξύ καταστάσεων και ενεργειών, αλλά υπάρχει και ένας βαθμός τυχαιότητας, το περιβάλλον είναι στοχαστικό. Τότε, ακόμα και όταν ο πράκτορας βρίσκεται σε συγκεκριμένη κατάσταση και εκτελεί μία συγκεκριμένη ενέργεια, δεν μπορεί να είναι βέβαιος για την ανταμοιβή που θα λάβει ή την επόμενη κατάσταση του περιβάλλοντος. Ένα παράδειγμα στοχαστικού περιβάλλοντος είναι το παιχνίδι του πόκερ, όπου οι κάρτες που θα αποκαλυφθούν στην επόμενη φάση του παιχνιδιού είναι τυχαίες.

#### Διακριτό ή συνεχές

Ένα περιβάλλον μπορεί να είναι διακριτό ή συνεχές ανάλογα με το πλήθος των δυνατών καταστάσεων

του. Όταν το πλήθος αυτό είναι πεπερασμένο, τότε το περιβάλλον είναι διακριτό. Για παράδειγμα, η τρίλιζα είναι ένα παιχνίδι με διακριτό περιβάλλον, καθώς οι δυνατές καταστάσεις του παιχνιδιού είναι πεπερασμένες και μάλιστα, λίγες.

Αντίθετα, όταν το πλήθος των δυνατών καταστάσεων είναι άπειρο, τότε το περιβάλλον είναι συνεχές. Ένα παράδειγμα συνεχούς περιβάλλοντος είναι το περιβάλλον της πλοιήγησης ενός ρομπότ, όπου η θέση, η ταχύτητα κι ο προσανατολισμός του ρομπότ μπορούν να λάβουν οποιαδήποτε τιμή σε ένα συνεχές πεδίο τιμών.

### Μονοπρακτορικό ή πολυπρακτορικό

Ανάλογα με το πλήθος των πρακτόρων που συμμετέχουν στο περιβάλλον, αυτό χαρακτηρίζεται ως μονοπρακτορικό ή πολυπρακτορικό. Τα παζλ αποτελούν παραδείγματα μονοπρακτορικού περιβάλλοντος, ενώ αντίθετα, αθλήματα όπως το ποδόσφαιρο είναι πολυπρακτορικά, καθώς συμμετέχουν πολλοί πράκτορες, οι οποίοι συνεργάζονται ή ανταγωνίζονται μεταξύ τους.

### Επεισοδιακό ή ακολουθιακό

Το περιβάλλον μπορεί να είναι επεισοδιακό ή ακολουθιακό ανάλογα με τον τρόπο με τον οποίο ο πράκτορας αλληλεπιδρά με αυτό. Σε επεισοδιακά περιβάλλοντα, η αλληλεπίδραση μεταξύ πράκτορα και περιβάλλοντος χωρίζεται σε επεισόδια, τα οποία τερματίζονται μετά από την επίτευξη του επιθυμητού στόχου ή μετά από ένα συγκεκριμένο αριθμό βημάτων. Για παράδειγμα το σκάκι μπορεί να θεωρηθεί επεισοδιακό περιβάλλον, καθώς κάθε παιχνίδι αποτελεί ένα επεισόδιο, το οποίο τελειώνει είτε μέσω ρουά ματ ή όταν τελειώσει ο χρόνος ενός παίκτη.

Αντίθετα, σε ακολουθιακά περιβάλλοντα, η αλληλεπίδραση μεταξύ πράκτορα και περιβάλλοντος δεν έχει φυσικό τέλος και ο πράκτορας συνεχίζει να αλληλεπιδρά με το περιβάλλον για αόριστο χρονικό διάστημα. Ένα παράδειγμα ακολουθιακού περιβάλλοντος είναι η πλοιήγηση ενός ρομπότ σε έναν άγνωστο χώρο.

### Με αραίες ή πυκνές ανταμοιβές

Η διάκριση εδώ αφορά τον ρυθμό με τον οποίο ο πράκτορας λαμβάνει ανταμοιβές από το περιβάλλον. Σε περιβάλλοντα με αραίες ανταμοιβές, ο πράκτορας λαμβάνει ανταμοιβή σπάνια, μόνο όταν επιτύχει έναν συγκεκριμένο στόχο. Για παράδειγμα στην τρίλιζα, ο πράκτορας επιβραβεύεται μόνο όταν καταφέρνει να κερδίσει το παιχνίδι.

Αντίθετα, σε περιβάλλοντα με πυκνές ανταμοιβές, ο πράκτορας λαμβάνει ανταμοιβή πιο συχνά, προσφέροντας του έτσι, πιο άμεση ανατροφοδότηση για την ποιότητα των ενεργειών του. Ένα παράδειγμα πυκνών ανταμοιβών είναι το κλασικό περιβάλλον εκπαίδευσης CartPole, οπου ο πράκτορας προσπαθεί να διατηρήσει σε ισορροπία μία όρθια ράβδο. Στο περιβάλλον αυτό, όσο η ράβδος παραμένει σε ισορροπία, ο πράκτορας λαμβάνει συνεχώς ανταμοιβές.

#### Ανταμοιβή

Σε κάθε ενέργεια του πράκτορα, το περιβάλλον του επιστρέφει μία τιμή, την ανταμοιβή. Πρόκειται για έναν αριθμό, ο οποίος μπορεί να είναι είτε αρνητικός (ο πράκτορας «τιμωρείται») ή θετικός (ο πράκτορας «επιβραβεύεται»). Ουσιαστικά, η ανταμοιβή δείχνει πόσο καλή ή κακή είναι η κατάσταση που βρίσκεται ο πράκτορας. Σκοπός του πράκτορα είναι να μεγιστοποιήσει την αθροιστική ανταμοιβή του σε βάθος χρόνου.

Η συνάρτηση η οποία υπολογίζει σε κάθε βήμα του πράκτορα την ανταμοιβή που λαμβάνει, ονομάζεται Συνάρτηση Ανταμοιβής (*Reward Function*). Η συνάρτηση ανταμοιβής μπορεί να είναι απλή, με τη λογική της επιβράβευσης ή τιμωρίας στο τέλος του παιχνιδιού, ή πιο πολύπλοκη, με περισσότερες επιβραβεύσεις και τιμωρίες, ώστε να οδηγήσει τον πράκτορα στην επιθυμητή συμπεριφορά. Για παράδειγμα, ας θεωρήσουμε το παιχνίδι της πλοϊγήσης σε έναν λαβύρινθο, όπου στόχος του πράκτορα είναι να βρει την έξοδο του. Μία απλή συνάρτηση ανταμοιβής θα μπορούσε να είναι η ανταμοιβή +1, όταν βρει την έξοδο και -1, εφόσον τελειώσει ο διαθέσιμος χρόνος του πράκτορα. Σε αυτήν την περίπτωση, η συνάρτηση ανταμοιβής είναι αραιή (*Sparse Rewards*). Αντίθετα, μία πιο πολύπλοκη συνάρτηση ανταμοιβής θα μπορούσε να είναι η ανταμοιβή +1000 όταν βρει την έξοδο, +10 όταν πλησιάζει σε αυτήν, -10 όταν απομακρύνεται από αυτήν και -1000 εφόσον τελειώσει ο διαθέσιμος χρόνος του πράκτορα. Η δεύτερη, αναλυτικότερη προσέγγιση ονομάζεται Διαμόρφωση Ανταμοιβής (*Reward Shaping*) και συνήθως οδηγεί σε καλύτερα αποτελέσματα, καθώς παρέχει περισσότερη ανάδραση στον πράκτορα, καθοδηγώντας τον προς τον τελικό στόχο. Ωστόσο, χρειάζεται περισσότερη προσοχή από τον σχεδιαστή του συστήματος, προκειμένου να αποφευχθούν μη επιθυμητές συμπεριφορές του πράκτορα. Για αυτό, κάποιες φορές είναι προτιμότερο να διατηρηθεί απλή η συνάρτηση ανταμοιβής, ακόμα κι αν αυτό σημαίνει μεγαλύτερο χρόνο εκπαίδευσης.

Η ορθή σχεδίαση μίας συνάρτησης ανταμοιβής είναι καθοριστική για την επιτυχία του πράκτορα στην εκπαίδευση. Ωστόσο, η διαδικασία αυτή αποδεικνύεται στην πράξη δύσκολη, καθώς περιέχει αρκετές προκλήσεις και απαιτεί προσεκτική ανάλυση του προβλήματος. Περισσότερες λεπτομέρειες σχετικά με τα προβλήματα που μπορεί να προκύψουν από τη σχεδίαση της συνάρτησης ανταμοιβής και τις πρακτικές που χρησιμοποιούνται για την αντιμετώπισή τους, δίνονται στο κεφάλαιο της εκπαίδευσης, στις υποενότητες [5.5.4](#) και [5.6.3](#) αντίστοιχα.

#### Κατάσταση

Η κατάσταση (*state*) αποτελεί το σύνολο των πληροφοριών που δέχεται ο πράκτορας από το περιβάλλον, σε μία ορισμένη χρονική στιγμή. Ουσιαστικά, πρόκειται για την κωδικοποίηση της οπτικής αναπαράστασης του περιβάλλοντος, σε μορφή πληροφοριών που μπορούν να δοθούν ως είσοδο στον πράκτορα. Για παράδειγμα, όταν οι άνθρωποι παίζουν ένα παιχνίδι, το οπτικό κανάλι είναι αυτό που λαμβάνει τις περισσότερες πληροφορίες για την κατάσταση του παιχνιδιού. Σε ένα

επιτραπέζιο παιχνίδι όπως η τρίλιζα, οι άνθρωποι βλέπουν την εικόνα του ταμπλό και με βάση αυτήν παίρνουν αποφάσεις. Όμως, ένας πράκτορας χρειάζεται δεδομένα σε μορφή αριθμών ως είσοδο για το νευρωνικό του δίκτυο. Συνεπώς, η κατάσταση πρέπει να κωδικοποιηθεί σε ένα διάνυσμα αριθμών. Για την περίπτωση της τρίλιζας θα μπορούσαμε να είχαμε ένα διάνυσμα 9 θέσεων, όπου κάθε μία συμβολίζει ένα τετράγωνο του ταμπλό και μπορεί να πάρει 3 διακριτές τιμές: 1 αν στη θέση υπάρχει Ο, -1 αν υπάρχει X και 0 αν είναι άδεια. Έτσι, το διάνυσμα  $[0, 0, 0, 0, -1, 0, 0, 0, 0]$  αντιστοιχεί σε ένα ταμπλό με X στη μεσαία θέση.

Στο παραπάνω παράδειγμα, το κάθε στοιχείο του διανύσματος κατάστασης μπορούσε να πάρει συγκεκριμένες, διακριτές τιμές. Έτσι, ο χώρος καταστάσεων που δημιουργείται είναι διακριτός (*discrete state space*). Υπάρχουν όμως περιπτώσεις, στις οποίες το κάθε στοιχείο του διανύσματος κατάστασης μπορεί να πάρει οποιαδήποτε τιμή σε ένα συνεχές διάστημα. Σε αυτές τις περιπτώσεις, ο χώρος καταστάσεων είναι συνεχής (*continuous state space*) και ο αριθμός των διαφορετικών καταστάσεων του πράκτορα είναι άπειρος. Για παράδειγμα, στην πλοήγηση ενός ρομπότ σε έναν άγνωστο χώρο, η κατάσταση του ρομπότ μπορεί να περιγραφεί από τη θέση του στο χώρο, την ταχύτητά του και τον προσανατολισμό του. Κάθε μία από αυτές τις παραμέτρους μπορεί να πάρει οποιαδήποτε τιμή σε ένα συνεχές διάστημα κι έτσι ο χώρος καταστάσεων είναι συνεχής. Ο χαρακτηρισμός του χώρου καταστάσεων ως διακριτός ή συνεχής είναι καθοριστικής σημασίας, διότι παίζει σημαντικό ρόλο στην επιλογή του αλγορίθμου εκπαίδευσης του πράκτορα.

Γενικά, είναι προτιμότερο οι πληροφορίες που δίνονται στον πράκτορα, να περιορίζονται μόνο στις χρήσιμες σε αυτόν για την επίτευξη του στόχου του. Αυτό έχει ως αποτέλεσμα τη μείωση της διαστασιμότητας του προβλήματος, κάτι που επιταχύνει την εκπαίδευση του πράκτορα. Ωστόσο, η περιορισμένη πληροφορία μπορεί να οδηγήσει σε ανεπαρκή εκπαίδευση του πράκτορα, καθώς αυτός δεν έχει την πλήρη εικόνα του περιβάλλοντος και μπορεί να χάσει σημαντικές πληροφορίες. Επομένως, η σχεδίαση της κατάστασης πρέπει να γίνει με προσοχή, ώστε να εξασφαλιστεί η ισορροπία μεταξύ της πληροφορίας και της διαστασιμότητας του προβλήματος.

## Ενέργεια

Ο όρος «ενέργεια» αντιπροσωπεύει μία δράση που μπορεί να πραγματοποιήσει ο πράκτορας στο περιβάλλον του. Σε κάθε χρονική στιγμή  $t$ , ο πράκτορας πρέπει να επιλέξει μία ενέργεια από το σύνολο των διαθέσιμων ενεργειών του. Οι ενέργειες μπορεί να είναι διακριτές ή συνεχείς. Όπως και πριν, στο παιχνίδι της τρίλιζας, ο αριθμός των δυνατών κινήσεων του πράκτορα είναι πεπερασμένος και άρα ο χώρος ενεργειών είναι διακριτός (*discrete action space*). Αντίθετα, στην πλοήγηση ενός ρομπότ σε έναν άγνωστο χώρο, δεν αρκεί το ρομπότ να επιλέξει να κινηθεί π.χ. γρήγορα προς τα δεξιά, αλλά απαιτείται μεγαλύτερη ακρίβεια στην κίνηση του. Έτσι, το ρομπότ επιλέγει συγκεκριμένη ταχύτητα και γωνία κίνησης, με αποτέλεσμα ο χώρος ενεργειών να είναι συνεχής (*continuous action space*). Η επιλογή των δυνατών ενεργειών του πράκτορα από τον σχεδιαστή του συστήματος είναι σημαντική,

### 3.3 Ενισχυτική Μάθηση

---

επειδή επηρεάζει την πολυπλοκότητα του προβλήματος και την επιλογή του αλγορίθμου εκπαίδευσης.

#### Πολιτική

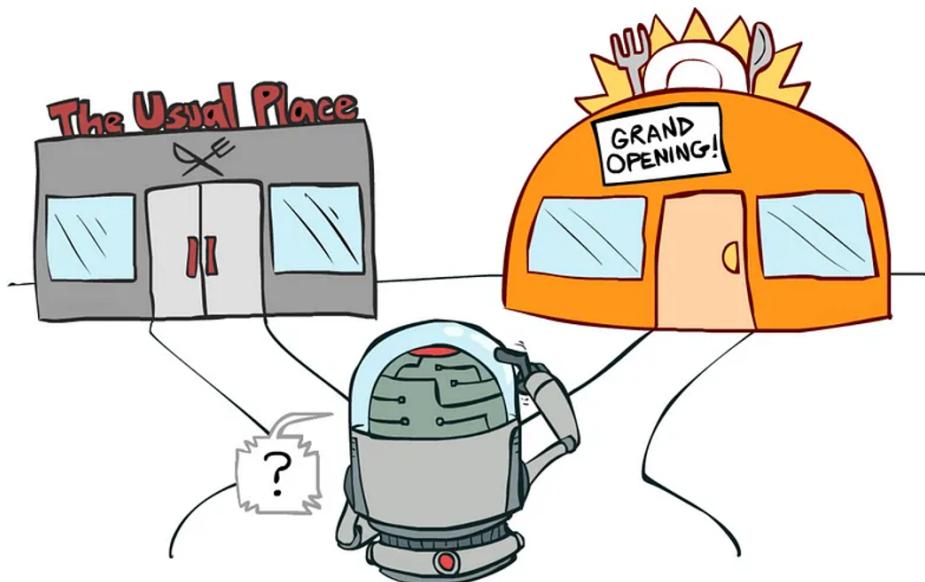
Η πολιτική περιγράφει τον τρόπο με τον οποίο ο πράκτορας επιλέγει την επόμενη ενέργειά του σε κάθε κατάσταση. Από τεχνικής άποψης, η πολιτική του πράκτορα αποτελεί απλώς μία αντιστοιχίση της κάθε κατάστασης σε μία ενέργεια ή, αν θέλουμε να την παρομοιάσουμε με τον τρόπο που λειτουργούν οι άνθρωποι, πρόκειται για το σκεπτικό το οποίο χρησιμοποιεί ο πράκτορας για τη λήψη αποφάσεων. Η πολιτική αυτή μπορεί να είναι ντετερμινιστική ή στοχαστική. Στην πρώτη περίπτωση, συμβολίζεται ως  $a_{t+1} = \pi(s_t)$ , δηλαδή η πολιτική αποτελεί την συνάρτηση που δέχεται ως είσοδο την κατάσταση του περιβάλλοντος και επιστρέφει την ενέργεια του πράκτορα. Τότε, όταν ο πράκτορας βρίσκεται στην ίδια κατάσταση, θα επιλέγει πάντα την ίδια ενέργεια. Στη δεύτερη περίπτωση, η πολιτική αντιστοιχίζει κάθε μία από τις δυνατές ενέργειες κάθε κατάστασης του πράκτορα, σε πιθανότητες. Συμβολίζεται ως  $\pi(a_i | s_t)$  και αντιπροσωπεύει την πιθανότητα να επιλέξει την ενέργεια  $a_i$  ενώ βρίσκεται στην κατάσταση  $s_t$ . Εξυπακούεται ότι το άθροισμα των πιθανοτήτων όλων των πιθανών ενεργειών σε μία κατάσταση ισούται με τη μονάδα. Προκειμένου να γίνει καλύτερα κατανοητός ο τρόπος λειτουργίας μίας στοχαστικής πολιτικής, ας θεωρησούμε το παράδειγμα του κλασικού arcade παιχνιδιού «Snake» (φιδάκι). Στο παιχνίδι αυτό, ο πράκτορας έχει 4 δυνατές ενέργειες: πάνω, κάτω, αριστερά και δεξιά. Σε μία συγκεκριμένη κατάσταση, η στοχαστική πολιτική θα αντιστοιχίζει σε κάθε μία από τις 4 δυνατές κινήσεις του πράκτορα, μία πιθανότητα. Για παράδειγμα, αν οι πιθανότητες είναι:  $P(\text{πάνω}) = 0.7$ ,  $P(\text{κάτω}) = 0.2$ ,  $P(\text{αριστερά}) = 0.05$  και  $P(\text{δεξιά}) = 0.05$ , τότε ο πράκτορας θα κινηθεί πάνω με πιθανότητα 0.7, κάτω με πιθανότητα 0.2 και αριστερά και δεξιά με πιθανότητα 0.05. Γενικά, οι στοχαστικές πολιτικές είναι πιο ευέλικτες από τις ντετερμινιστικές, καθώς επιτρέπουν στον πράκτορα να εξερευνήσει το περιβάλλον του και να ανακαλύψει νέες στρατηγικές. Ωστόσο, η επιλογή της στοχαστικής πολιτικής απαιτεί προσεκτική ανάλυση, καθώς μπορεί να οδηγήσει σε ανεπιθύμητες συμπεριφορές του πράκτορα.

Πλέον, γίνεται κατανοητό ότι η πολιτική που αναπτύσσει ο πράκτορας είναι ο παράγοντας που εν τέλει, καθορίζει την επιτυχία ή αποτυχία της εκπαίδευσης του. Το ζητούμενο είναι ο πράκτορας να αναπτύξει τη βέλτιστη πολιτική, η οποία θα τον οδηγήσει στην επίτευξη του στόχου του με τον πιο αποδοτικό τρόπο. Ωστόσο, η ανάπτυξη της βέλτιστης πολιτικής αποτελεί στην πράξη μία πρόκληση, καθώς απαιτεί την εξερεύνηση του περιβάλλοντος και την ανακάλυψη των στρατηγικών που θα οδηγήσουν στην επίτευξη του στόχου.

Έχοντας εξηγήσει την έννοια της πολιτικής, είμαστε πλέον σε θέση να παρουσιάσουμε το βασικότερο ίσως, πρόβλημα της ενισχυτικής μάθησης, το δίλημμα της εξερεύνησης έναντι της εκμετάλλευσης.

### Δίλημμα Εξερεύνησης - Εκμετάλλευσης

Το δίλημμα της εξερεύνησης έναντι της εκμετάλλευσης (*exploration vs exploitation dilemma*) αποτελεί ένα από τα πιο δυσεπίλητα προβλήματα στο πεδίο της ενισχυτικής μάθησης. Το δίλημμα αναφέρεται στην ισορροπία μεταξύ της εξερεύνησης, δηλαδή της ανακάλυψης νέων περιοχών και της εκμετάλλευσης, δηλαδή της χρήσης της υπάρχουσας γνώσης. Έτσι, προκύπτει το ερώτημα: πρέπει ο πράκτορας να συνεχίσει να εφαρμόζει τις ενέργειες που γνωρίζει ότι λειτουργούν (εκμετάλλευση) ή να δοκιμάσει νέες ενέργειες, προκειμένου να ανακαλύψει νέες στρατηγικές που ίσως είναι ακόμα πιο αποδοτικές (εξερεύνηση); Για παράδειγμα, πρέπει κάποιος να παραγγέλνει πάντα το ίδιο, γνωστό πιάτο στο αγαπημένο του εστιατόριο ή να δοκιμάσει κάτι καινούριο και διαφορετικό, με την ελπίδα να ανακαλύψει κάτι καλύτερο; Το παράδειγμα αυτό παρουσιάζεται και παραστατικά, στην *Εικόνα 3.6*.



**Εικόνα 3.6.** Το δίλημμα Εξερεύνησης - Εκμετάλλευσης (Parkinson 2019).

### Ιδανική εκτέλεση διαδικασίας

Προκειμένου να κατανοήσουμε σε βάθος το δίλημμα αυτό και τις επιπλοκές που έχει στην εκπαίδευση του πράκτορα, θα ξεκινήσουμε εξηγώντας τη διαδικασία που θα έπρεπε, ιδανικά, να ακολουθηθεί, ώστε ο πράκτορας να αναπτύξει την βέλτιστη πολιτική. Στη συνέχεια, θα αναδείξουμε τα προβλήματα που ανακύπτουν, τα οποία δυσκολεύουν την ομαλή εξέλιξη της παραπάνω διαδικασίας.

Στα πρώτα στάδια της εκπαίδευσης, επιθυμούμε ο πράκτορας να εξερευνήσει το περιβάλλον. Με τον όρο «εξερεύνηση», εννοούμε ο πράκτορας να δοκιμάσει τυχαίες ενέργειες σε διαφορετικές κατάστασεις, προκειμένου να αποκτήσει εμπειρία και να καταλάβει ποιές καταστάσεις είναι καλές και ποιές όχι. Δεν μας ενδιαφέρει ακόμα η απόδοση του πράκτορα, αλλά η ανάπτυξη μίας

### 3.3 Ενισχυτική Μάθηση

---

ισχυρής βάσης γνώσης. Είναι σημαντικό άλλωστε, να έχουμε κατά νου πως οι πράκτορες τεχνητής νοημοσύνης δεν έχουν όλες τις προηγούμενες γνώσεις κι εμπειρίες που έμεις, οι άνθρωποι, έχουμε αναπτύξει, ήδη από μικρή ηλικιά. Για παράδειγμα, στο περιβάλλον εκπαίδευσης αυτής της εργασίας, στόχος του πράκτορα είναι η στάθμευση ενός αυτοκινητού σε μία ελεύθερη θέση. Εάν ο χειριστής του αυτοκινήτου ήταν άνθρωπος, αμέσως μόλις έβλεπε το περιβάλλον εκπαίδευσης (τον χώρο πάρκινγκ), θα είχε ήδη αρκετές χρήσιμες γνώσεις, όπως το πως κινείται ένα αυτοκίνητο, το ότι ο στόχος του είναι να παρκάρει το αυτοκίνητο στην ελεύθερη θέση ή το ότι η σύγκρουση με άλλα, σταθμευμένα αυτοκίνητα είναι -το λιγότερο- ανεπιθύμητη. Ωστόσο, ο πράκτορας δεν γνωρίζει ακόμα τίποτα από αυτά και πρέπει να τα ανακαλύψει μόνος του, εξερευνώντας το περιβάλλον και μαθαίνοντας από τις ανταμοιβές που πάίρνει. Έτσι, προς το παρόν, επιθυμούμε παραδείγματος χάριν, ο πράκτορας να συγκρούεται με άλλα αυτοκίνητα, ώστε να καταλάβει ότι αυτό δεν είναι επιθυμητό και να το αποφεύγει στα επόμενα στάδια της εκπαίδευσης του.

Μετά από ένα επαρκές διάστημα εξερεύνησης, ο πράκτορας πρέπει να περάσει στο στάδιο της εκμετάλλευσης. Στο στάδιο αυτό, ο πράκτορας δεν επιλέγει πια τυχαίες ενέργειες, αλλά αξιοποιεί τη γνώση που έχει ήδη αποκτήσει, ώστε να επιλέγει σε κάθε κατάσταση, την καλύτερη ενέργεια. Η μετάβαση από το ένα στάδιο στο άλλο πρέπει να γίνει σταδιακά, δηλαδή επιθυμούμε ο πράκτορας να ελαττώσει βαθμιαία την τυχαιότητα των κινήσεων του, έως ότου τη μηδενίσει και βασίζεται εξ ολοκλήρου στην πολιτική του για τη λήψη αποφάσεων. Μόνο με αυτόν τον τρόπο, θα καταφέρει ο πράκτορας να ανακαλύψει τη βέλτιστη συμπεριφορά. Για παράδειγμα, στο περιβάλλον αυτόματης στάθμευσης, επιθυμούμε ο πράκτορας να καταλάβει νωρίς στην εκπαίδευση, ότι το να πλησιάζει την ελεύθερη θέση είναι καλό. Έτσι, όσο βασίζεται όλο και περισσότερο στην πολιτική του για τη λήψη αποφάσεων, τόσο θα επιλέγει ενέργειες που θα τον κινούν πιο κοντά στην ελεύθερη θέση. Ταυτόχρονα όμως, διατηρεί έναν βαθμό τυχαιότητας, ο οποίος αν και μειωμένος, τον ωθεί να εξακολουθεί να εξερευνεί νέες καταστάσεις και στρατηγικές. Έτσι, θα ανακαλύψει κάποτε τη μεγάλη επιβράβευση της στάθμευσης, θα εξερευνήσει διαφορετικούς τρόπους για να φτάνει σε αυτήν και στο τέλος της εκπαίδευσης, όταν θα αξιοποιεί αποκλειστικά την πολιτική του, θα επιλέγει πάντα τον βέλτιστο τρόπο στάθμευσης.

#### Προβλήματα κατά την εκτέλεση της διαδικασίας

Για την επιτυχία των εκπαιδεύσεων ενισχυτικής μάθησης, είναι κρίσιμη η ισορροπία μεταξύ της εξερεύνησης και της εκμετάλλευσης, δηλαδή ο ρυθμός με τον οποίο ο πράκτορας μειώνει την τυχαιότητα των ενεργειών του. Η ισορροπία αυτή είναι στην πράξη πολύ δύσκολο να επιτευχθεί, καθώς δίνεται περισσότερη βαρύτητα στο ένα από τα δύο στάδια.

Συγκεκριμένα, εάν ο πράκτορας εξερευνήσει πολύ λίγο τον χώρο, τότε δεν θα αποκτήσει επαρκείς εμπειρίες και δεν θα ανακαλύψει τον βέλτιστο τρόπο για την επίτευξη του στόχου του - ή μπορεί και να μην ανακαλύψει καν τον στόχο του. Για να γίνει καλύτερα κατανοητό αυτό, ας επιστρέψουμε στο παράδειγμα της αυτόματης στάθμευσης κι ας εξετάσουμε μία τέτοια συμπεριφορά που συνέβη

πολλές φορές στις εκπαιδεύσεις των πρακτόρων. Η επιλογή τυχαίων ενεργειών στα πρώτα στάδια της εκπαίδευσης, έχει ως αποτέλεσμα ο πράκτορας να συγκρούεται συχνά με άλλα αυτοκίνητα. Αν η εξερεύνηση σταματήσει πρόωρα, τώρα ο πράκτορας θα αρχίσει να εκμεταλλεύεται τη γνώση που έμαθε. Η γνώση αυτή, είναι μόνο ότι οι συγκρούσεις είναι ανεπιθύμητες, κι έτσι θα τις αποφεύγει. Ωστόσο, ο πράκτορας δεν πρόλαβε να μάθει πως η στάθμευση οδηγεί σε μεγάλη επιβράβευση κι έτσι, δεν επιχειρεί ποτέ να παρκάρει, αλλά κάνει απλώς κύκλους γύρω από την πίστα του παιχνιδιού. Τότε, λέμε πως ο πράκτορας έχει υιοθετήσει μία υποβέλτιστη πολιτική (*suboptimal policy*), ή έχει παγιδευτεί σε τοπικό μέγιστο (*local maximum*). Η δεύτερη έκφραση αναφέρεται στη συνάρτηση ανταμοιβής, καθώς γνωρίζουμε πως στόχος του πράκτορα είναι να τη μεγιστοποιεί, δηλαδή να πάρνει ως ανταμοιβή το ολικό μέγιστο της. Όμως, όταν η εξερεύνηση του πράκτορα είναι ανεπαρκής, αυτός παίρνει ένα τοπικό μέγιστο της συνάρτησης ανταμοιβής, το οποίο είναι μικρότερο από το ολικό.

Από την άλλη, εάν ο χρόνος εξερεύνησης είναι πολύ μεγάλος, αυξάνεται ο χρόνος εκτέλεσης του πειράματος, επειδή ο πράκτορας αφιερώνει σημαντικό χρόνο σε μη χρήσιμες καταστάσεις. Ακόμα, υπάρχει ο κίνδυνος η απόδοση του πράκτορα να είναι ιδιαίτερα ασταθής, καθώς συνεχίζει να δοκιμάζει νέες ενέργειες αντί να βελτιώνει και να εξελίσσει τις γνωστές του στρατηγικές. Έτσι, μπορεί να μην καταφέρει να αναπτύξει μία σταθερή πολιτική, η οποία να τον οδηγεί στον στόχο του. Μάλιστα, επιθυμούμε ο πράκτορας να εξερευνεί «προς τη σωστή κατεύθυνση». Αυτό σημαίνει πως δεν θέλουμε ο πράκτορας να σπαταλάει μεγάλο χρόνο εκπαίδευσης εξερευνώντας μη χρήσιμες καταστάσεις, όπως για το παράδειγμα της αυτόματης στάθμευσης, διαφορετικούς τρόπους να συγκρούεται με άλλα αυτοκίνητα, αλλά θέλουμε να εξερευνεί καταστάσεις χρήσιμες, που τον φέρνουν όλο και πιο κοντά στον τελικό του στόχο. Συνεπώς, η επιτυχία του πράκτορα στο στάδιο της εκμετάλλευσης, εξαρτάται από την ποιότητα της εμπειρίας που έχει αποκτήσει κατά τη διάρκεια της εξερεύνησης. Εάν η εμπειρία αυτή είναι ανεπαρκής ή ανεπιθύμητη, τότε ο πράκτορας θα αντιμετωπίσει δυσκολίες στην επίτευξη του στόχου του.

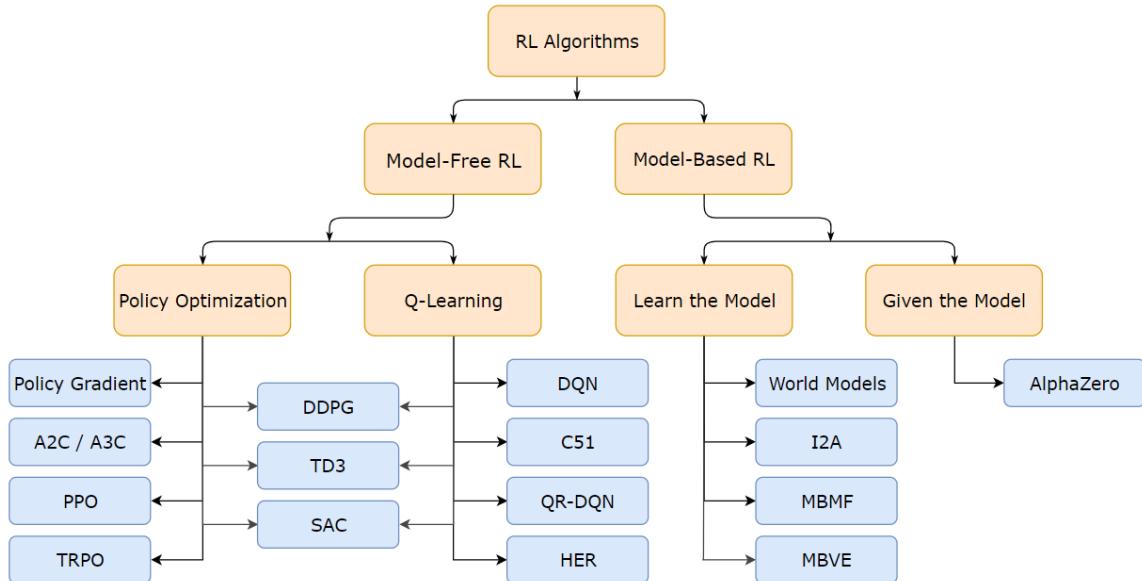
Μέχρι και σήμερα, δεν υπάρχει κάποια γενικά αποδεκτή λύση στο πρόβλημα της εξερεύνησης έναντι της εκμετάλλευσης. Έχουν προταθεί οριμένες τεχνικές για την επίτευξη της ζητούμενης ισορροπίας, όπως ο αλγόριθμος  $\epsilon$ -greedy και η κανονικοποίηση της εντροπίας (*entropy regularization*), τις οποίες θα εξετάσουμε σε επόμενες ενότητες, όμως καμία δεν εγγυάται την επίτευξη της βέλτιστης πολιτικής.

### 3.3.3 Κατηγορίες αλγορίθμων

Έχοντας κατανοήσει τις βασικές αρχές της ενισχυτικής μάθησης, μπορούμε πλέον να μελετήσουμε τους επικρατέστερους αλγορίθμους που χρησιμοποιούνται για την εκπαίδευση πρακτόρων. Στην υποενότητα αυτή, θα προβούμε σε μία επισκόπηση των διαφορετικών κατηγοριών αλγορίθμων και σε μεταγενέστερες υποενότητες, θα εξετάσουμε τους αλγορίθμους που χρησιμοποιήθηκαν στα πλαίσια αυτής της εργασίας.

### 3.3 Ενισχυτική Μάθηση

Μία συνοπτική ταξινόμηση των δημοφιλέστερων αλγορίθμων ενισχυτικής μάθησης παρουσιάζεται στην *Εικόνα 3.7*.



**Εικόνα 3.7.** Ταξινόμηση αλγορίθμων ενισχυτικής μάθησης (OpenAI 2018).

Παρατηρούμε πως μία πρώτη διάκριση των αλγορίθμων ενισχυτικής μάθησης γίνεται σε αλγορίθμους *Model-Free* και *Model-Based*.

#### Model-Free vs Model-Based

Η διάσπαση των αλγορίθμων στις δύο βασικές αυτές κατηγορίες εξαρτάται από την πρόσβαση ή μη του πράκτορα στο μοντέλο του περιβάλλοντος. Με τον όρο «μοντέλο του περιβάλλοντος» εννοούμε μία συνάρτηση που προβλέπει τις μεταβάσεις καταστάσεων και τις ανταμοιβές. Οι αλγόριθμοι model based απαιτούν την ύπαρξη ενός μοντέλου του περιβάλλοντος, ενώ οι model free όχι.

#### Model Based

Το μεγάλο πλεονέκτημα της γνώσης του μοντέλου έγκειται στο γεγονός ότι ο πράκτορας γνωρίζει όλες τις πιθανές ενέργειες του και τα αποτελέσματα αυτών. Επομένως, έχει τη δυνατότητα να σχεδιάσει την επόμενη κίνηση του, επιλέγοντας την καλύτερη από τις διαθέσιμες ενέργειες. Με τον τρόπο αυτό, οι αλγόριθμοι model based χρησιμοποιούν το μοντέλο για την ανάπτυξη μίας βέλτιστης πολιτικής. Όταν αυτή η προσέγγιση λειτουργεί, μπορεί να οδηγήσει σε σημαντική βελτίωση της αποδοτικότητας σε σχέση με αλγορίθμους που δεν διαθέτουν μοντέλο.

Ωστόσο, το κύριο πρόβλημα αυτών των αλγορίθμων έγκειται στη δημιουργία του μοντέλου. Συγκεκριμένα, το μοντέλο αυτό είτε προυπάρχει και είναι διαθέσιμο στον πράκτορα, είτε πρέπει να

δημιουργηθεί από τον ίδιο. Στην πλειοψηφία των προβλημάτων ενισχυτικής μάθησης, το μοντέλο του περιβάλλοντος είναι άγνωστο και πρέπει να αναπτυχθεί από τον πράκτορα κατά την εκπαίδευση του, μέσω των εμπειριών του. Αυτό παρουσιάζει αρκετές προκλήσεις, καθώς η πολυπλοκότητα του πραγματικού κόσμου, καθιστά δύσκολη τη δημιουργία ενός ακριβού μοντέλου. Κάθε απόκλιση μεταξύ του μοντέλου και του πραγματικού κόσμου (*model bias*), μπορεί να οδηγήσει σε μειωμένη απόδοση του πράκτορα, κατά την εφαρμογή του στον πραγματικό κόσμο. Συνολικά, η εκμάθηση του μοντέλου είναι δύσκολη και μπορεί να αποτύχει, ακόμα και αν αφιερωθεί πολύς χρόνος και υπολογιστική ισχύς στην ανάπτυξή του.

### Model Free

Οι αλγόριθμοι model free δεν απαιτούν γνώση ενός μοντέλου του περιβάλλοντος, αλλά στηρίζονται αποκλειστικά στην αλληλεπίδραση του πράκτορα με το περιβάλλον. Επομένως, ο πράκτορας χρησιμοποιεί την εμπειρία του, για την ανάπτυξη της πολιτικής του. Το μειονέκτημα αυτής της προσέγγισης, είναι ότι συνήθως απαιτείται περισσότερος χρόνος εκπαίδευσης, προεκιμένου ο πράκτορας να αναπτύξει μία αποδοτική πολιτική. Ωστόσο, η απουσία ανάπτυξης ενός μοντέλου του περιβάλλοντος, καθιστά τους αλγορίθμους model free πιο εύκολους στην υλοποίηση και τη ρύθμιση. Επιπλέον, οι αλγορίθμοι αυτοί αποδεικνύονται πιο σταθεροί και αξιόπιστοι, καθώς δεν επηρεάζονται από τυχόν σφάλματα και αποκλίσεις του μοντέλου. Έτσι, οι μέθοδοι χωρίς μοντέλο είναι πιο δημοφιλείς κι έχουν αναπτυχθεί και δοκιμαστεί περισσότερο από τις μεθόδους με μοντέλο.

Για τους λόγους που αναλύθηκαν παραπάνω, όλοι οι αλγόριθμοι που υλοποιήθηκαν στα πλαίσια αυτής της εργασίας ανήκουν στην κατηγορία των model free αλγορίθμων. Ως εκ τούτου, στη συνέχεια θα αναλύσουμε μόνο τις υποκατηγορίες των model free αλγορίθμων.

### Κατηγορίες Model Free Αλγορίθμων

Υπάρχουν δύο βασικές προσεγγίσεις στους model free αλγορίθμους: οι αλγόριθμοι Εκτίμησης Αξίας (*Value Estimation*) και οι αλγόριθμοι Βελτιστοποίησης Πολιτικής (*Policy Optimization*). Υπάρχει όμως κι ένας τρίτος τύπος model free αλγορίθμων, οι αλγόριθμοι Δράστη-Κριτή (*Actor-Critic*), οι οποίοι συνδυάζουν στοιχεία από τις δύο προηγούμενες προσεγγίσεις. Κάθε ένα από αυτούς τους τύπους έχει τα δικά του πλεονεκτήματα και μειονεκτήματα, και είναι κατάλληλος για διαφορετικούς τύπους προβλημάτων. Οι 3 αυτοί τύποι φαίνονται στο κάτω αριστερά τμήμα της Εικόνας 3.7<sup>1</sup> και εξετάζονται στη συνέχεια.

#### Αλγόριθμοι Εκτίμησης Αξίας

Οι αλγόριθμοι Εκτίμησης Αξίας (*Value Estimation*) χωρίζονται σε δύο υποκατηγορίες, τους αλγορίθμους Εκτίμησης Αξίας Κατάστασης (*State Value Estimation*) και τους αλγορίθμους Εκτίμησης

<sup>1</sup>Οι αλγόριθμοι Εκτίμησης Αξίας παρουσιάζονται στην εικόνα ως Q-Learning αλγόριθμοι, ενώ οι αλγόριθμοι που βρίσκονται μεταξύ των δύο βασικών προσεγγίσεων αποτελούν τους αλγορίθμους Δράστη-Κριτή.

### 3.3 Ενισχυτική Μάθηση

---

Αξίας Ζεύγους Κατάστασης-Ενέργειας (*State-Action Value Estimation*). Η λογική πίσω από αυτούς τους αλγορίθμους όμως παραμένει η ίδια και στις 2 περιπτώσεις και αναλύεται στη συνέχεια.

Οι αλγόριθμοι Εκτίμησης Αξίας Καταστάσεων επικεντρώνονται στην εκμάθηση της αξίας των καταστάσεων. Η βασική ιδέα είναι η ανάπτυξη μίας συνάρτησης αξίας (*value function*), η οποία συμβολίζεται ως  $V_\pi(s)$  και θα εκτιμάει την ποιότητα μίας κατάστασης. Ο όρος «αξία» ή «ποιότητα» μίας κατάστασης αναφέρεται στην αναμενόμενη αθροιστική ανταμοιβή που θα λάβει ο πράκτορας, ξεκινώντας από την κατάσταση αυτή και ακολουθώντας έπειτα την πολιτική που έχει αναπτύξει.

Κατά παρόμοιο τρόπο, οι αλγόριθμοι Εκτίμησης Αξίας Ζευγών Κατάστασης-Ενέργειας εστιάζουν στην εκμάθηση της αξίας των ζευγών κατάστασης-ενέργειας. Η συνάρτηση αξίας σε αυτή την περίπτωση συμβολίζεται ως  $Q_\theta(s, a)$  και εκτιμά την ποιότητα της ενέργειας  $a$  στην κατάσταση  $s$ , δηλαδή την αναμενόμενη αθροιστική ανταμοιβή που θα λάβει ο πράκτορας, εάν επιλέξει την ενέργεια  $a$  στην κατάσταση  $s$  και ακολουθήσει έπειτα την πολιτική που έχει αναπτύξει. Οι ενημερώσεις της συνάρτησης αξίας γίνονται *off-policy*, το οποίο σημαίνει ότι κάθε ενημέρωση μπορεί να χρησιμοποιήσει δεδομένα που συλλέχθηκαν σε οποιοδήποτε σημείο της εκπαίδευσης, ανεξάρτητα από το πώς εξερευνούσε τότε ο πράκτορας το περιβάλλον. Χαρακτηριστικό παράδειγμα αυτής της προσέγγισης είναι ο αλγόριθμος Q-Learning και για αυτό, οι συγκεκριμένοι αλγόριθμοι συχνά αναφέρονται και ως Q-Learning αλγόριθμοι.

Τα μεγαλύτερο πλεονέκτημα των αλγορίθμων Εκτίμησης Αξίας είναι η απλότητά τους. Ακόμα, είναι ιδιαίτερα αποτελεσματικοί σε περιβάλλοντα με διακριτούς χώρους καταστάσεων και ενεργειών. Στα περιβάλλοντα αυτά, είναι σημαντικά πιο αποδοτικοί στον χρόνο εκπαίδευσης σε σχέση με τις άλλες κατηγορίες αλγορίθμων, επειδή μπορούν να επαναχρησιμοποιήσουν τα ίδια δεδομένα πολλές φορές. Όμως, οι αλγόριθμοι Εκτίμησης Αξίας βελτιστοποιούν μόνο έμμεσα την απόδοση του πράκτορα, μέσω της ενημέρωσης της συνάρτησης αξίας. Επίσης, παρουσιάζουν δυσκολία στην αντιμετώπιση μεγάλων ή συνεχών χώρων καταστάσεων και ενεργειών, αφού απαιτούν την αποθήκευση μίας τιμής για κάθε ζεύγος κατάστασης-ενέργειας. Έτσι, συνολικά, τείνουν να είναι λιγότερο σταθεροί από τους αλγορίθμους των άλλων δύο κατηγοριών.

### Αλγόριθμοι Βελτιστοποίησης Πολιτικής

Οι αλγόριθμοι Βελτιστοποίησης Πολιτικής (*Policy Optimization*) μαθαίνουν απευθείας μία πολιτική  $\pi_\theta$ , που μεγιστοποιεί την αναμενόμενη αθροιστική ανταμοιβή, χωρίς να απαιτούν την ανάπτυξη συνάρτησης αξιών. Αντίθετα, επικεντρώνονται στη ρύθμιση των παραμέτρων  $\theta$  της πολιτικής, προκειμένου να βελτιστοποίησουν την απόδοση τους. Αυτή η ρύθμιση πραγματοποιείται *on-policy*, δηλαδή κάθε ενημέρωση χρησιμοποιεί μόνο δεδομένα που συλλέχθηκαν ενώ ο πράκτορας ενεργούσε σύμφωνα με την πιο πρόσφατη έκδοση της πολιτικής του. Όπως αναφέρθηκε και νωρίτερα, η πολιτική μπορεί να είναι ντετερμινιστική ή στοχαστική. Μία ντετερμινιστική πολιτική συμβολίζεται ως  $a_{t+1} = \pi_\theta(s_t)$  και υποδηλώνει ότι σε συγκεκριμένη κατάσταση, θα επιλέγεται κάθε φορά η ίδια ενέργεια. Αντίθετα, μία στοχαστική πολιτική συμβολίζεται ως  $\pi_\theta(a_i|s_t)$  και εισάγει ένα βαθμό

τυχαιότητας στη λήψη αποφάσεων από τον πράκτορα, αφού η επιλεγμένη ενέργεια προκύπτει από μία κατανομή πιθανοτήτων. Παραδείγματα αλγορίθμων αυτής της κατηγορίας αποτελούν οι κλασικοί αλγόριθμοι TRPO και PPO.

Το μεγαλύτερο πλεονέκτημα που έχουν οι αλγόριθμοι της κατηγορίας βελτιστοποίησης πολιτικής είναι ότι βελτιστοποιούν άμεσα την πολιτική τους, δηλαδή τον στόχο της εκπαίδευσης. Αυτό το γεγονός τους καθιστά σταθερούς και αξιόπιστους. Ακόμα, αξιοσημείωτη είναι η απόδοση τους σε περιβάλλοντα με συνεχείς χώρους καταστάσεων ή και ενεργειών. Ωστόσο, συχνά απαιτούν μεγάλο χρόνο εκπαίδευσης για την επίτευξη σταθερής μάθησης. Επιπλέον, απαιτούν πολύ προσεκτική ρύθμιση των παραμέτρων τους, επειδή είναι επιρρεπείς σε τοπικά ακρότατα.

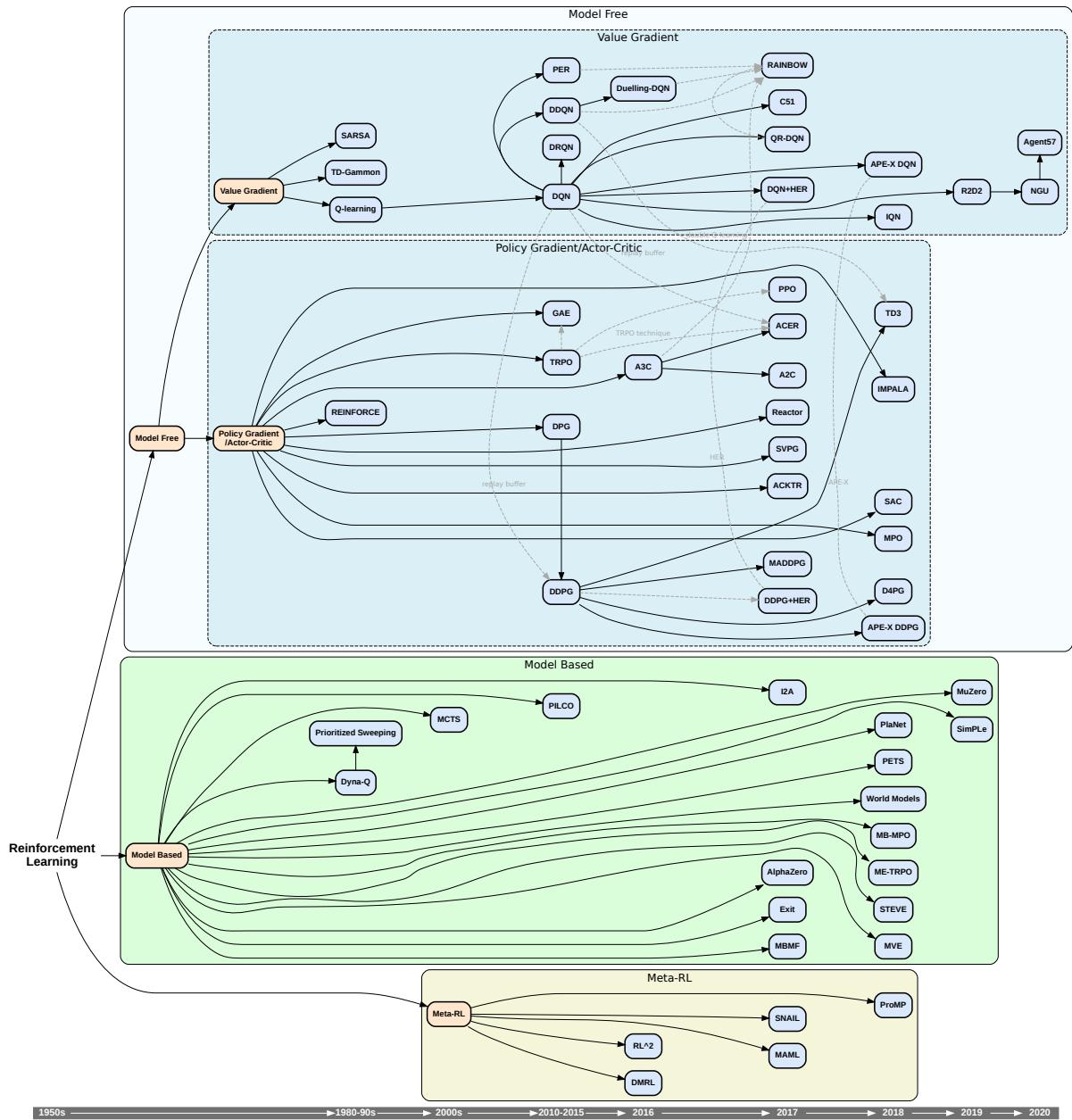
#### Αλγόριθμοι Δράστη - Κριτή

Οι αλγόριθμοι Δράστη-Κριτή (*Actor-Critic*) συνδυάζουν στοιχεία από τις δύο προηγούμενες κατηγορίες. Έτσι, προσπαθούν να εκμεταλλευτούν τα πλεονεκτήματα αλλά και να αποφύγουν τις αδυναμίες της κάθε προσέγγισης. Αποτελούνται από δύο ξεχωριστά νευρωνικά δίκτυα: ένα δίκτυο-δράστη και ένα δίκτυο-κριτή. Ο δράστης αποφασίζει ποια ενέργεια να πάρει ο πράκτορας και έπειτα ο κριτής αξιολογεί την επιλεγμένη ενέργεια, με την εκτίμηση της συνάρτησης αξίας. Στη συνέχεια, ο κριτής παρέχει στον δράστη ανατροφοδότηση, για να βελτιώσει την πολιτική του. Μάλιστα, ο κριτής βοηθά στη μείωση της διακύμανσης στις ενημερώσεις της πολιτικής, οδηγώντας σε πιο σταθερή μάθηση. Παραδείγματα αλγορίθμων αυτής της κατηγορίας αποτελούν οι SAC και TD3.

Το μεγαλύτερο πλεονέκτημα των αλγορίθμων δράστη-κριτή είναι πως αξιοποιούν τόσο τις τεχνικές της εκτίμησης αξίας, όσο και της βελτιστοποίησης πολιτικής. Έτσι, η διαδικασία αυτή, επιτυγχάνει συνήθως, μία καλή ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης, οδηγώντας σε πιο σταθερή και αποτελεσματική μάθηση. Χρησιμοποιείται ευρύτερα σε προβλήματα που περιλαμβάνουν πολύπλοκα περιβάλλοντα και μεγάλους χώρους καταστάσεων. Ωστόσο, οι αλγόριθμοι αυτοί είναι πιο πολύπλοκοι στην υλοποίηση και τη ρύθμιση, λόγω της αλληλεπίδρασης μεταξύ του δράστη και του κριτή. Μπορούν ακόμα να υποφέρουν από αστάθεια ή και απόκλιση, εάν ο κριτής παρέχει κακές εκτιμήσεις.

Τέλος, αξίζει να γίνει ειδική μνεία σε μία άλλη, λιγότερο γνωστή απόπειρα ταξινομήσης του μεγάλου πλήθους των αλγορίθμων ενισχυτικής μάθησης από τον (Prijono 2020), η οποία παρουσιάζεται στην *Εικόνα 3.8*.

### 3.3 Ενισχυτική Μάθηση



**Εικόνα 3.8.** Εκτενέστερη ταξινόμηση αλγορίθμων ενισχυτικής μάθησης (Prijono 2020).

Στην εικόνα αυτή, τα βέλη που συνδέουν δύο αλγορίθμους, υποδηλώνουν ότι ο ένας αλγόριθμος αποτελεί βελτίωση του άλλου. Συγκεκριμένα, οι συνεχείς γραμμές υποδηλώνουν ισχυρή σύνδεση μεταξύ των δύο αλγορίθμων, ενώ οι διακεκομμένες γραμμές υποδηλώνουν πιο ασθενή σύνδεση. Έτσι, καταλαβαίνουμε για παράδειγμα, πως οι αλγόριθμοι PPO, TD3 που χρησιμοποιηθήκαν στην παρούσα εργασία, αποτελούν βελτιώσεις των αλγορίθμων TRPO και DDPG αντίστοιχα. Επιπλέον, στο κάτω μέρος της εικόνας, παρουσιάζεται το χρονολόγιο της δημοσίευσης των αλγορίθμων, προκειμένου να γίνει κατανοητή η εξέλιξη των αλγορίθμων στο χρόνο. Μάλιστα, στον σύνδεσμο που υπάρχει στη βιβλιογραφία, υπάρχει ένα πλήρες αποθετήριο, όπου για κάθε αλγόριθμο μπορεί κάνεις να βρει την αντίστοιχη δημοσίευση, καθώς και άλλες χρήσιμες πληροφορίες. Για αυτό, προτρέπω τους ενδιαφερόμενους αναγνώστες να επισκεφτούν και να χρησιμοποιήσουν το συγκεκριμένο αποθετήριο, για να περιηγηθούν γρήγορα και εύκολα στον χώρο των αλγορίθμων ενισχυτικής μάθησης.

### 3.3.4 Ο αλγόριθμος *Q-learning*

Ο αλγόριθμος *Q-Learning* προτάθηκε από τον Chris Watkins το 1989, στη διδακτορική του διατριβή (Watkins 1989). Είναι ένας από τους πιο διαδεδομένους αλγορίθμους ενισχυτικής μάθησης και αποτέλεσε τη βάση για πολλές εξελίξεις στον τομέα. Ανήκει στην κατηγορία των model free αλγορίθμων και πιο συγκεκριμένα, στην κατηγορία των αλγορίθμων εκτίμησης αξίας ζευγών κατάστασης-ενέργειας.

#### Μεθοδολογία αλγορίθμου

Όπως κι οι υπόλοιποι αλγόριθμοι της κατηγορίας αξίας ζευγών κατάστασης-ενέργειας, ο *Q-Learning* επικεντρώνεται στην εκμάθηση της αξίας των ζευγών κατάστασης-ενέργειας. Η συνάρτηση αξίας που αναπτύσσει συμβολίζεται ως  $Q(s, a)$ , όπου το «*Q*» αντιπροσωπεύει την ποιότητα (*Quality*) της ενέργειας *a* στην κατάσταση *s*, δηλαδή πόσο χρήσιμη είναι η συγκεκριμένη ενέργεια στη συγκεκριμένη κατάσταση, στο να μεγιστοποιήσει τις μελλοντικές ανταμοιβές. Συνήθως, χρησιμοποιείται ένας πίνακας (*Q Table*) για την αποθήκευση των τιμών *Q* της συνάρτησης  $Q(s, a)$ , όπου κάθε γραμμή αντιστοιχεί σε μία κατάσταση και κάθε στήλη σε μία ενέργεια, οπως φαίνεται στην *Εικόνα 3.9*.

	Actions			
	$A_1$	$A_2$	...	$A_M$
$S_1$	$Q(S_1, A_1)$	$Q(S_1, A_2)$		$Q(S_1, A_M)$
$S_2$	$Q(S_2, A_1)$	$Q(S_2, A_2)$		$Q(S_2, A_M)$
:			$\ddots$	$\ddots$
$S_N$	$Q(S_N, A_1)$	$Q(S_N, A_2)$	...	$Q(S_N, A_M)$

**Εικόνα 3.9.** Πίνακας  $Q$  για την αποθήκευση των τιμών της συνάρτησης  $Q(s, a)$  (Baeldung 2023).

Το πιο σημαντικό κομμάτι του αλγορίθμου  $Q$ -Learning είναι η κατασκευή της συνάρτησης  $Q(s, a)$ . Η κατασκευή αυτή γίνεται μέσω της επαναληπτικής ενημέρωσης των τιμών  $Q$ .

Αρχικά, ας δούμε πως προκύπτει μία τιμή  $Q$  για ένα ζεύγος κατάστασης-ενέργειας. Η τιμή  $Q$  για το ζεύγος  $(s, a)$  υπολογίζεται κατά τη διάρκεια της εκπαίδευσης από την εξίσωση 3.2 (εξίσωση Bellman):

$$Q(s, a) = R(s, a) + \gamma \cdot \max_a Q(s', a) \quad (3.2)$$

Η εξίσωση αυτή, δηλώνει ότι η αξία του ζεύγους  $(s, a)$  είναι ίση με την ανταμοιβή που προκύπτει από την εκτέλεση της ενέργειας  $a$  στην κατάσταση  $s$  (συμβολίζεται ως  $R(s, a)$ ), συν την αναμενόμενη ανταμοιβή που προκύπτει εκτελώντας την καλύτερη ενέργεια  $a$  σε όλες τις επόμενες καταστάσεις  $s'$  (συμβολίζεται ως  $\max_a Q(s', a)$ ), πολλαπλασιασμένη επί τον συντελεστή  $\gamma$ . Ο συντελεστής  $\gamma$ , ονομάζεται παράγοντας έκπτωσης (*discount factor*), παίρνει τιμές στο διάστημα  $[0, 1]$  και χρησιμοποιείται για τον καθορισμό της σημασίας των μελλοντικών ανταμοιβών. Συγκεκριμένα, οι μελλοντικές ανταμοιβές είναι λιγότερο πολύτιμες από τις τρέχουσες ανταμοιβές, και για αυτό ο παράγοντας  $\gamma$  τις μετριάζει. Όταν παίρνει τιμές κοντά στο 0, τότε ο πράκτορας συμπεριφέρεται πιο κοντόφθαλμα, ενώ όσο η τιμή του παράγοντα πλησιάζει το 1, τόσο ο πράκτορας προσπαθεί να μεγιστοποιήσει το μακροπρόθεσμο κέρδος του.

Ωστόσο, όπως αναφέραμε, η ενημέρωση των τιμών  $Q$  γίνεται επαναληπτικά κατά τη διάρκεια της εκπαίδευσης. Έτσι, στην αρχή της εκπαίδευσης, οι τιμές  $Q$  όλων των ζευγών κατάστασης-ενέργειας αρχικοποιούνται σε μία προκαθορισμένη τιμή (συνήθως 0). Έπειτα, όταν ο πράκτορας επισκέπτεται μία κατάσταση  $s$ , εκτελεί μία ενέργεια  $a$  και λαμβάνει μία ανταμοιβή  $R(s, a)$ . Τότε, η τιμή  $Q$  του ζεύγους  $(s, a)$  ενημερώνεται σύμφωνα με την εξίσωση 3.3:

$$Q^{new}(s, a) = (1 - \alpha) \cdot Q^{old}(s, a) + \alpha \left[ R(s, a) + \gamma \cdot \max_a Q(s', a) \right] \quad (3.3)$$

Επειδή αυτή η εξίσωση χρησιμοποιεί τη διαφορά χρησιμότητας μεταξύ διαδοχικών καταστάσεων (κι επομένως, διαδοχικών χρόνων), ονομάζεται και Εξίσωση Ενημέρωσης Χρονικών Διαφορών (*Temporal Difference Update Rule*). Παρατηρούμε ότι η εξίσωση TD Update Rule χρησιμοποιεί την εξίσωση Bellman. Συγκεκριμένα, η νέα τιμή  $Q$  του ζεύγους  $(s, a)$  συμβολίζεται ως  $Q^{new}(s, a)$  και προκύπτει από:

- τον όρο  $(1 - \alpha) \cdot Q^{old}(s, a)$ , που αντιπροσωπεύει την παλιά τιμή  $Q$  του ζεύγους  $(s, a)$ , δηλαδή αυτήν που ήδη υπήρχε στον πίνακα  $Q$ , πολλαπλασιασμένη με τον συντελεστή  $(1 - \alpha)$  και
- τον όρο  $\alpha [R(s, a) + \gamma \cdot \max_a Q(s', a)]$ , που αντιπροσωπεύει την τιμή  $Q$  του ζεύγους  $(s, a)$  που μόλις υπολογίστηκε από την εξίσωση Bellman, πολλαπλασιασμένη με τον συντελεστή  $\alpha$ .

Ο συντελεστής  $\alpha$  ονομάζεται ρυθμός μάθησης (*learning rate*) και καθορίζει σε ποιό βαθμό, οι νεότερες, πιο πρόσφατες πληροφορίες αντικαθιστούν τις παλιές, δηλαδή καθορίζει τον ρυθμό με τον οποίο ο πράκτορας μαθαίνει. Ο ρυθμός μάθησης παίρνει κι αυτός τιμές στο διάστημα  $[0, 1]$  και αποτελεί μία ακόμα, κρίσιμη υπερπαράμετρο που πρέπει να ρυθμιστεί με προσοχή, καθώς επηρεάζει σημαντικά τη σύγκλιση του αλγορίθμου. Εάν πάρει την τιμή 0, τότε ο πίνακας  $Q$  δεν ενημερώνεται καθόλου (δηλαδή ο πράκτορας δεν μαθαίνει τίποτα καινούργιο), ενώ εάν πάρει την τιμή 1, τότε ο πίνακας  $Q$  ενημερώνεται πλήρως από την τιμή που προκύπτει από την εξίσωση Bellman, δηλαδή ο πράκτορας λαμβάνει υπόψη μόνο τις πιο πρόσφατες πληροφορίες. Συνήθως, οι τιμές του συντελεστή αυτού μειώνονται κατά τη διάρκεια της εκπαίδευσης, ώστε να επιτευχθεί μία καλή ισορροπία μεταξύ της εκμάθησης και της σταθερότητας του αλγορίθμου.

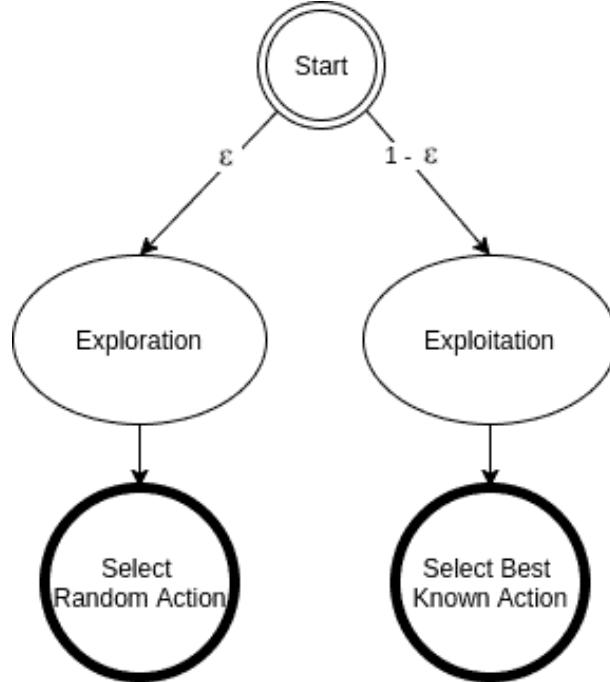
Τέλος, αφού ο πράκτορας κατασκευάσει τη συνάρτηση  $Q(s, a)$ , μέσα από πολλές ενημερώσεις των τιμών  $Q$ , τότε μπορεί να ενεργεί βέλτιστα, μέσα από τον υπολογισμό  $a = \arg \max_a Q(s, a)$ , δηλαδή επιλέγοντας απλώς την ενέργεια με τη μεγαλύτερη τιμή  $Q$  σε κάθε κατάσταση.

### Η τεχνική $\varepsilon$ -greedy

Ένα σημαντικό κομμάτι της παραπάνω διαδικασίας, είναι το πως ο πράκτορας επιλέγει την ενέργεια που θα εκτελέσει σε κάθε κατάσταση, όταν βρίσκεται ακόμα στο στάδιο της εκπαίδευσης. Πρόκειται για το δίλημμα εξερεύνησης-εκμετάλλευσης, το οποίο αναλύθηκε στην υποενότητα 3.3.2. Ο αλγόριθμος  $Q$ -Learning αντιμετωπίζει το πρόβλημα αυτό, χρησιμοποιώντας την τεχνική  $\varepsilon$ -greedy.

### 3.3 Ενισχυτική Μάθηση

Η τεχνική  $\varepsilon$ -greedy είναι μία διαδεδομένη μέθοδος για την εξισορρόπηση της εξερεύνησης και της εκμετάλλευσης. Είναι εύκολη στην υλοποίηση, αλλά ταυτόχρονα αποδίδει εξίσου καλά με άλλες, πιο πολύπλοκες μεθόδους. Η τεχνική αυτή παρουσιάζεται υπό μορφή διαγράμματος ροής (flowchart) στην Εικόνα 3.10.



Εικόνα 3.10. Η τεχνική  $\varepsilon$ -greedy (Baeldung 2023).

Σημαντικό ρόλο στην τεχνική αυτή παίζει η παράμετρος  $\varepsilon$ , η οποία λαμβάνει τιμές στο διάστημα  $[0,1]$  και συμβολίζει την πιθανότητα ο πράκτορας να εξερευνήσει το περιβάλλον. Συγκεκριμένα, κατά τη διάρκεια της εκπαίδευσης, όταν ο πράκτορας βρίσκεται σε μία κατάσταση  $s$ , επιλέγει με πιθανότητα  $\varepsilon$ , τυχαία μία από τις διαθέσιμες ενέργειες (εξερεύνηση), ενώ με πιθανότητα  $1 - \varepsilon$ , επιλέγει την ενέργεια που έχει τη μεγαλύτερη τιμή  $Q(s, a)$  στον πίνακα εκείνη τη στιγμή (εκμετάλλευση).

Συνήθως, επιλέγεται η παράμετρος  $\varepsilon$  να μειώνεται σταδιακά κατά την εκπαίδευση κι έτσι, η τεχνική ονομάζεται και ως φθίνουσα (decaying)  $\varepsilon$ -greedy. Με τον τρόπο αυτό, δίνεται μεγαλύτερη έμφαση στην εξερεύνηση στα πρώτα στάδια της εκπαίδευσης και στην εκμετάλλευση στα τελευταία. Με άλλα λόγια, ο πράκτορας δρα σε μεγάλο βαθμό τυχαία στην αρχή της εκπαίδευσης, όπου η τιμή του  $\varepsilon$  είναι υψηλή, ενώ όταν η τιμή του  $\varepsilon$  φτάσει το 0, ο πράκτορας γίνεται «άπληστος» (greedy), δηλαδή επιλέγει πάντα την βέλτιστη ενέργεια. Ο ρυθμός με τον οποίο μειώνεται η τιμή της πιθανότητας, είναι συνήθως είτε γραμμικός είτε εκθετικός και εξαρτάται από τον αριθμό των καταστάσεων και των ενεργειών.

Θεωρητικά, σε άπειρο χρόνο εκπαίδευσης του αλγορίθμου  $Q$ -Learning, η πολιτική του πράκτορα θα αντιστοιχεί στη βέλτιστη (Melo 2001). Έτσι, δεν θα είχε τότε, νόημα, η εξερεύνηση στο περιβάλλον.

Ωστόσο, σε πραγματικά προβλήματα ενισχυτικής μάθησης, ο πράκτορας εκπαιδεύεται για πεπερασμένο αριθμό βημάτων, κι επομένως η τιμή  $\varepsilon$  δεν τείνει ποτέ στο μηδέν. Αντίθετα, τίθεται ένα κατώτατο όριο, για παράδειγμα η τιμή 0.001. Με αυτόν τον τρόπο, εξασφαλίζεται ότι ο πράκτορας δεν θα σταματήσει ποτέ, πλήρως, την εξερεύνηση, μιας και η πολιτική του θα έχει πάντα περιθώρια βελτίωσης.

Δεν υπάρχει συγκεκριμένος κανόνας για τον καθορισμό του ρυθμού μείωσης του  $\varepsilon$ , ούτε για το κατώτατο όριο του. Αντίθετα, ορίζονται από τον σχεδιαστή του συστήματος, μετά από πειραματισμό και δοκιμές. Είναι σημαντική η προσεκτική ρύθμιση αυτών των παραμέτρων, διότι επηρεάζουν σημαντικά τη σύγκλιση του αλγορίθμου.

Η μεθοδολογία του αλγορίθμου *Q-Learning* δίνεται παράκατω υπό μορφή ψευδοκώδικα (*pseudocode*).

---

### Q-Learning Algorithm

---

```

1: Initialize  $Q(s, a)$  to 0 for all  $a \in A$  in each  $s \in S$ 
2: Initialize learning rate  $\alpha \in (0, 1]$ 
3: Initialize discount factor  $\gamma \in [0, 1]$ 
4: Initialize exploration rate  $\epsilon \in [0, 1]$ 
5: while not converged do
6:    $s \leftarrow s_0$ 
7:   while  $s$  not terminal do
8:     Observe current state  $s$ 
9:     if explore() then
10:        $a \leftarrow$  random action
11:     else
12:        $a \leftarrow \arg \max_a Q(s, a)$ 
13:     end if
14:     Take action  $a$ , observe reward  $R(s, a)$  and next state  $s'$ 
15:     Update Q-value:
16:       
$$Q(s, a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \left[ R(s, a) + \gamma \cdot \max_a Q(s', a) \right]$$

17:   end while
18: end while

```

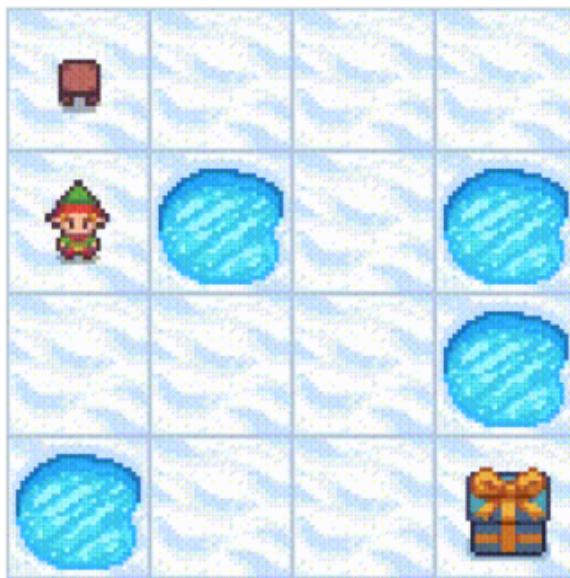
---

Συνολικά, η διαδικασία που ακολουθεί ο αλγόριθμος *Q-Learning* για την κατασκευή της συνάρτησης  $Q$ , θα γίνει καλύτερα κατανοητή με ένα παράδειγμα. Για αυτό, θα θεωρησούμε το παιχνίδι Frozen Lake της βιβλιοθήκης OpenAI Gymnasium<sup>2</sup>, που αποτελεί ένα κλασικό περιβάλλον εφαρμογής του αλγορίθμου *Q-Learning*.

<sup>2</sup>Η βιβλιοθήκη OpenAI Gymnasium αποτελεί μία από τις πιο δημοφιλείς βιβλιοθήκες για την εφαρμογή αλγορίθμων ενισχυτικής μάθησης σε περιβάλλοντα προσομοίωσης. Περιλαμβάνει μία μεγάλη ποικιλία από περιβάλλοντα, όπως το Frozen Lake, το CartPole, παιχνίδια Atari κ.ά. Επιπλέον, παρέχει μία εύχρηστη διεπαφή (API) για την αλληλεπίδραση με τα περιβάλλοντα, καθώς και πολλές χρήσιμες συναρτήσεις για την εκπαίδευση των αλγορίθμων.

### Παράδειγμα: Το παιχνίδι Frozen Lake

Στο παιχνίδι της παγωμένης λίμνης, η πίστα είναι ενα πλέγμα 4x4 και στόχος του πράκτορα είναι να φτάσει από την αρχή της πίστας (τετράγωνο πάνω αριστερά) στον στόχο του (τετράγωνο κάτω δεξιά), αποφεύγοντας τις τρύπες. Αν ο πράκτορας πέσει σε μία τρύπα, τότε το επεισόδιο τερματίζει και ο πράκτορας χάνει. Ο πράκτορας μπορεί να κινηθεί προς τα πάνω, κάτω, αριστερά και δεξιά. Ένα στιγμιότυπο του παιχνιδιού φαίνεται στην Εικόνα 3.11.



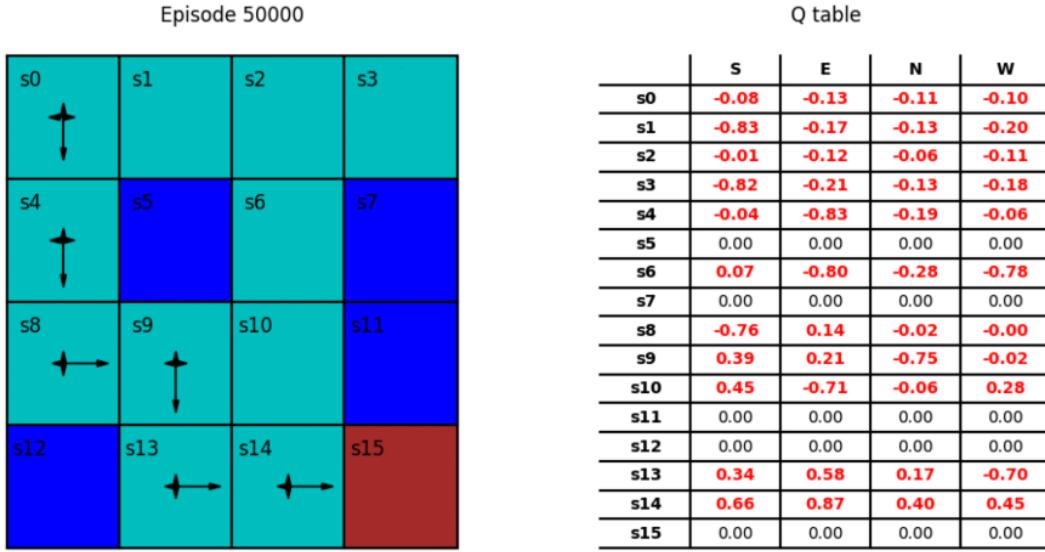
**Εικόνα 3.11.** Το παιχνίδι Frozen Lake της βιβλιοθήκης OpenAI Gymnasium.

Μοντελοποιώντας το παιχνίδι σε πρόβλημα ενισχυτικής μάθησης, μπορούμε να θεωρήσουμε ως κατάσταση του περιβάλλοντος, τη θέση του πράκτορα εντός του πλέγματος. Έτσι, υπάρχουν 16 δυνατές καταστάσεις. Επίσης, όπως αναφέραμε προηγουμένως, ο πράκτορας έχει 4 δυνατές ενέργειες. Η συνάρτηση ανταμοιβής μπορεί να εκφραστεί ως εξής (αραιές ανταμοιβές):

- +1 εάν ο πράκτορας φτάσει το στόχο
- -1 εάν ο πράκτορας πέσει σε τρύπα
- 0 σε όλες τις άλλες περιπτώσεις

Στην αρχή της εκπαίδευσης, οι τιμές του πίνακα  $Q$  αρχικοποιούνται στο 0. Ο πράκτορας επιλέγει ενέργειες χρησιμοποιώντας την τεχνική  $\varepsilon$ -greedy και ενημερώνει σε κάθε βήμα την αντίστοιχη τιμή  $Q$  του ζεύγους κατάστασης-ενέργειας, χρησιμοποιώντας την εξίσωση TD Update Rule. Μετά από 50000 επεισόδια, παρατηρείται πως το ποσοστό επιτυχίας του πράκτορα προσεγγίζει το 100%. Τότε, η εκπαίδευση σταματάει και ξεκινάει η αξιολόγηση του πράκτορα, όπου τίθεται η τιμή 0 στην παράμετρο  $\varepsilon$ , ώστε να επιλέγει ο πράκτορας πάντα την βέλτιστη ενέργεια, σύμφωνα με την πολιτική

του. Στην *Εικόνα 3.12* παρουσιάζεται η πολιτική που ανέπτυξε ο πράκτορας (αριστερά), καθώς και η τελική μορφή του πίνακα  $Q$  (δεξιά).



Policy (left) according to Q-table (right)

**Εικόνα 3.12.** Αποτελέσματα εκπαίδευσης στο παιχνίδι Frozen Lake (Szymanski 2018).

Τα βέλη στο πλέγμα αριστερά αντιστοιχούν στην ενέργεια, που επιλέγει στην αντίστοιχη κατάσταση ο πράκτορας. Όπως αποδεικνύεται κι από τον πίνακα δεξιά, η ενέργεια που επιλέγεται κάθε φορά, είναι αυτή με τη μεγαλύτερη τιμή  $Q$ . Ακόμα, αξίζει να σημειωθεί πως οι μόνες καταστάσεις με μηδενικές τιμές  $Q$  είναι οι τερματικές καταστάσεις, αφού σε αυτές ο πράκτορας δεν εκτελεί καμία ενέργεια, καθώς το παιχνίδι τελειώνει. Τελικά, η διαδρομή που σχηματίζεται από τις επιλεγμένες ενέργειες, οδηγεί τον πράκτορα από την αφετηρία στον στόχο, κι έτσι η εκπαίδευση θεωρείται επιτυχής.

### Αδυναμίες και πιθανές λύσεις

Ο αλγόριθμος  $Q$ -Learning αποτελεί έναν από τους θεμελιώδεις και πιο γνωστούς αλγορίθμους στον χώρο της ενισχυτικής μάθησης. Παρόλα αυτά, η χρήση του τα τελευταία χρόνια έχει μειωθεί σημαντικά, κάτι το οποίο οφείλεται στις σημαντικές αδυναμίες που εμφανίζει. Ωστόσο, πριν αναλύσουμε αυτές, ας επισημάνουμε πρώτα τα πλεονεκτήματά του αλγορίθμου, τα οποία τον κατέστησαν τόσο δημοφιλή:

- **Εύκολη υλοποίηση:** Η απλή φύση του αλγορίθμου καθιστά την υλοποίησή του αρκετά εύκολη, χωρίς να χρειάζεται η χρήση κάποιας βιβλιοθήκης μηχανικής μάθησης.
- **Μικρές απαιτήσεις σε υπολογιστικούς πόρους:** μπορεί να εκτελεστεί σε συστήματα με μικρή

### 3.3 Ενισχυτική Μάθηση

---

υπολογιστική ισχύ, σε αντίθεση με τους πιο μοντέρνους αλγορίθμους, που συνηθώς απαιτούν ειδικούς πόρους, όπως εξελιγμένες μονάδες επεξεργασίας γραφικών υπολογισμών (GPU).

- Κατάλληλος για προβλήματα με **μακροπρόθεσμα αποτελέσματα**: Χάρη στην ενημέρωση των τιμών  $Q$  με την εξίσωση TD Update Rule, ο αλγόριθμος είναι ικανός να αντιμετωπίσει προβλήματα με μακροπρόθεσμα αποτελέσματα, τα οποία είναι ιδιαίτερα απαιτητικά.

Στη συνέχεια, ας εξετάσουμε τις σημαντικότερες αδυναμίες του αλγορίθμου  $Q$ -Learning:

- Ακατάλληλος για **μεγάλους χώρους καταστάσεων και ενεργειών**: Σε προβλήματα με συνεχή χώρο καταστάσεων ή και ενεργειών, ο αλγόριθμος  $Q$ -Learning είναι πρακτικά, ανεφάρμοστος. Αυτό συμβαίνει, διότι ο αλγόριθμος απαιτεί την αποθήκευση των τιμών  $Q$  σε έναν πίνακα. Σε περιβάλλοντα όμως με μεγάλους χώρους καταστάσεων και ενεργειών, το πλήθος των διαθέσιμων ζευγών κατάστασης-ενέργειας είναι τεράστιο, με αποτέλεσμα ο πίνακας  $Q$  να γίνεται υπερβολικά μεγάλος και να απαιτεί μεγάλα ποσά μνήμης. Τα προβλήματα αυτά, που εμφανίζονται σε χώρους μεγάλων καταστάσεων είναι γνωστά και με το όνομα «**κατάρα της διάστασης**» (*curse of dimensionality*), ένας όρος που δόθηκε από τον Richard E. Bellman.
- **Έλλειψη γενίκευσης**: Η απλή μέθοδος  $Q$ -Learning δεν είναι ικανή να γενικεύσει τη γνώση που αποκτά. Με άλλα λόγια, ο πράκτορας δεν μπορεί να πάρει αποφάσεις για καταστάσεις που δεν έχει συναντήσει ξανά, ανεξάρτητα από την ομοιότητα της νέας αυτής κατάστασης με άλλες που έχει ήδη συναντήσει. Επομένως, σε προβλήματα με μεγάλο χώρο καταστάσεων, ακόμα και αν είχαμε τους απαραίτητους υπολογιστικούς πόρους για την αποθήκευση του πίνακα  $Q$ , ο χρόνος που θα χρειαζόταν για να εξερευνήσει ο πράκτορας κάθε ένα ζεύγος κατάστασης-ενέργειας, θα ήταν απαγορευτικά μεγάλος. Τελικά, η αδυναμία γενίκευσης της γνώσης αποτελεί άλλον έναν παράγοντα, για τον οποίο δεν είναι εφικτή η αποτελεσματική λειτουργία του αλγορίθμου σε πραγματικά προβλήματα.

Γίνεται πλέον κατανοητό, ότι η χρήση του αλγορίθμου  $Q$ -Learning σε πραγματικά προβλήματα είναι περιορισμένη, λόγω των παραπάνω αδυναμιών. Ωστόσο, υπάρχουν κάποιες προτάσεις για την αντιμετώπιση των προβλημάτων του αλγορίθμου. Οι κυριότερες από αυτές -και οι οποίες δοκιμάστηκαν στο περιβάλλον αυτόματης στάθμευσης- είναι οι εξής:

- **Διακριτοποίηση**: Μία τεχνική για τη μείωση του χώρου καταστάσεων και ενεργειών είναι η διακριτοποίηση των πιθανών τιμών. Για παράδειγμα, στο περιβάλλον αυτόματης στάθμευσης, αν η ταχύτητα του αυτοκινητού παίρνει συνεχείς τιμές στο διάστημα [0, 100], μπορεί να κβαντοποιηθεί στις τιμές [0, 1, 2, 3, 4], όπου το 0 αντιστοιχεί στην τιμή 0-20, το 1 στην τιμή 21-40 κ.ο.κ. Αν πραγματοποιηθεί αυτή η διαδικασία, σε κάθε ένα χαρακτηριστικό του χώρου καταστάσεων, τότε το πλήθος των στοιχείων του πίνακα  $Q$  μειώνεται σημαντικά. Έτσι, μπορεί να γίνει εφικτή η εφαρμογή του αλγορίθμου  $Q$ -Learning. Βέβαια, αν η διακριτοποίηση γίνει σε υπερβολικό βάθμο, μπορεί να προκαλέσει την έλλειψη ακρίβειας στην αναπαράσταση των καταστάσεων και να οδηγήσει σε χαμηλές επιδόσεις του πράκτορα.

- **Προσέγγιση συνάρτησης:** Μία άλλη τεχνική για την αντιμετώπιση του προβλήματος της μεγάλης διάστασης του πίνακα  $Q$  είναι η χρήση συναρτήσεων προσέγγισης. Συγκεκριμένα, μπορούν να χρησιμοποιηθούν νευρωνικά δίκτυα για να προσεγγίζουν τη συνάρτηση  $Q(s, a)$  και να εκτιμούν τις τιμές  $Q$ , χωρίς να χρειάζεται αυτές να αποθηκεύονται σε πίνακα. Έτσι, τα νευρωνικά δίκτυα είναι ικανά να γενικεύουν τη γνώση που αποκτούν, δηλαδή να παίρνουν αποφάσεις, για καταστάσεις που δεν έχουν συναντήσει ξανά αυτούσιες, κάτι που ο απλός πίνακας  $Q$  δεν μπορεί να κάνει. Η χρήση νευρωνικών δικτύων στον αλγόριθμο  $Q$ -Learning οδηγεί στη δημιουργία του αλγορίθμου Deep  $Q$ -Network (DQN). Ο αλγόριθμος αυτός αποτελεί μία από τις πιο δημοφιλείς εκδοχές του αλγορίθμου  $Q$ -Learning, καθώς είναι ικανός να αντιμετωπίσει προβλήματα με μεγάλους, συνεχείς χώρους καταστάσεων.

Η μέθοδος της προσέγγισης συνάρτησης αποτελεί σήμερα, την πιο συνηθισμένη λύση στο πρόβλημα της διαστασιμότητας. Με τον τρόπο αυτό, εισερχόμαστε στο πεδίο της Βαθιάς Ενισχυτικής Μάθησης, το οποίο περιγράφεται στην επόμενη ενότητα.

## 3.4 Βαθιά Ενισχυτική Μάθηση

### 3.4.1 Ορισμός και Χαρακτηριστικά

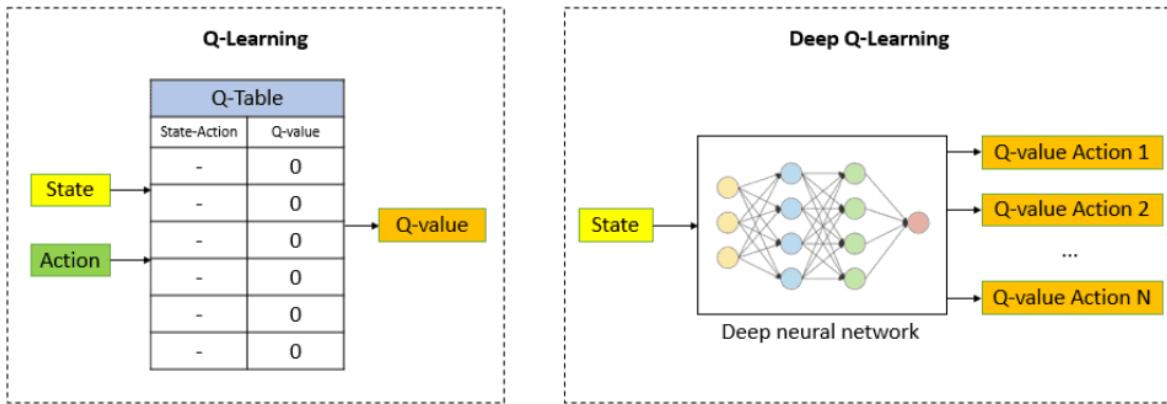
Τα προβλήματα της σύγχρονης εποχής χαρακτηρίζονται από υψηλή πολυπλοκότητα και μεγάλο πλήθος διαφορετικών καταστάσεων και ενεργειών. Η μεγάλη διαστασιμότητα των προβλημάτων αυτών, έχει θέσει σημαντικούς περιορισμούς στην εφαρμογή των αλγορίθμων της ενισχυτικής μάθησης. Συγκεκριμένα, σε τέτοιες περιπτώσεις, η εκπαίδευση και η ικανοποιητική εξερεύνηση του περιβάλλοντος καθίσταται χρονοβόρα, κοστοβόρα και εν τέλει, απαγορευτική από παραδοσιακούς αλγορίθμους ενισχυτικής μάθησης.

Η λύση δίνεται μέσω της μίμησης ενός βιολογικού μηχανισμού, των νευρωνικών δικτύων, και της ενσωμάτωσης τους στους αλγορίθμους ενισχυτικής μάθησης. Έτσι, δημιουργήθηκε ένα νέο επιστημονικό πεδίο, αυτό της Βαθιάς Ενισχυτικής Μάθησης. Επομένως, η Βαθιά Ενισχυτική Μάθηση (*Deep Reinforcement Learning*) αποτελεί απλά την ενοποίηση της Βαθιάς Μάθησης (*Deep Learning* - χρήση βαθιών νευρωνικών δικτύων) με την Ενισχυτική Μάθηση (*Reinforcement Learning*).

Μία σύγκριση της κλασικής ενισχυτικής μάθησης με τη βαθιά ενισχυτική μάθηση παρουσιάζεται στην *Εικόνα 3.13*, για την περίπτωση του αλγορίθμου  $Q$ -Learning.

Παρατηρώντας την *Εικόνα 3.13* γίνεται σαφής η λογική της βαθιάς ενισχυτικής μάθησης καθώς και τα πλεονεκτήματα που προκύπτουν από την χρήση νευρωνικών δικτύων. Συγκεκριμένα, στον κλασικό αλγόριθμο  $Q$ -Learning, όλες οι τιμές  $Q$  των ζευγών κατάστασης ενέργειας αποθηκεύονται σε έναν πίνακα. Ο πράκτορας ανατρέχει στον πίνακα αυτό, για να επιλέξει σε κάθε κατάσταση, την ενέργεια με τη μεγαλύτερη αξία. Αντίθετα, ο αλγόριθμος βαθιάς ενισχυτικής μάθησης DQN, αντικαθιστά τον

### 3.4 Βαθιά Ενισχυτική Μάθηση



Εικόνα 3.13. Κλασική Ενισχυτική Μάθηση εναντίον Βαθιάς Ενισχυτικής Μάθησης (Quang 2024).

προηγούμενο πίνακα με ένα βαθύ νευρωνικό δίκτυο. Το δίκτυο αυτό, εκτιμά σε κάθε κατάσταση, τις τιμές  $Q$  για όλες τις δυνατές ενέργειες, ώστε να επιλέξει ο πράκτορας τη βέλτιστη ενέργεια. Επομένως, αποφεύγεται η αποθήκευση των τιμών  $Q$ , ενώ επιτυγχάνεται η γενίκευση σε άγνωστες καταστάσεις.

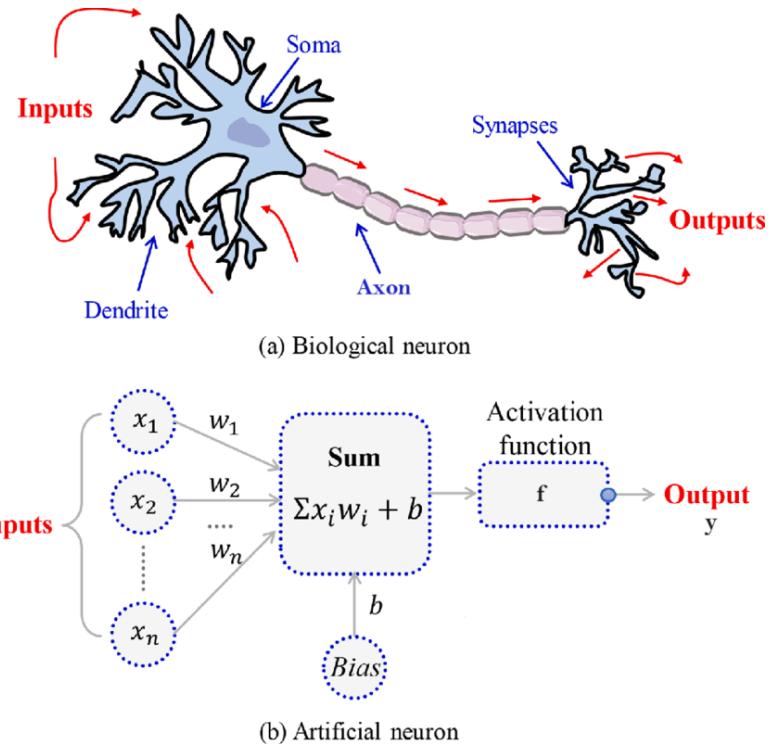
Τα τεχνητά νευρωνικά δίκτυα αποτελούν το αντικείμενο της επόμενης υποενότητας, όπου αρχικά συγκρίνονται με τα βιολογικά νευρωνικά δίκτυα, ενώ στη συνέχεια παρουσιάζονται οι βασικές αρχές λειτουργίας τους.

#### 3.4.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (*Artificial Neural Networks - ANNs*) είναι υπολογιστικά μοντέλα εμπνευσμένα από τη δομή και τη λειτουργία των βιολογικών νευρωνικών δικτύων που βρίσκονται στον ανθρώπινο εγκέφαλο. Αποτελούνται από συνδεδεμένα επίπεδα κόμβων (ή *νευρώνων*) που συνεργάζονται για να επεξεργαστούν και να μάθουν από τα δεδομένα.

#### Σύγκριση με Βιολογικά Νευρωνικά Δίκτυα

Όπως προαναφέρθηκε, τα τεχνητά νευρωνικά δίκτυα είναι σχεδιασμένα ώστε να μιμούνται τη λειτουργία των βιολογικών νευρωνικών δικτύων. Προκειμένου να κατανοήσουμε πώς λειτουργούν τα νευρωνικά δίκτυα, ας εξετάσουμε πρώτα πώς λειτουργούν τα βιολογικά νευρωνικά δίκτυα. Ο ανθρώπινος εγκέφαλος αποτελείται από δισεκατομμύρια κύτταρα που ονομάζονται νευρώνες, οι οποίοι επικοινωνούν μεταξύ τους μέσω ηλεκτρικών και χημικών σημάτων. Η Εικόνα 3.14 δείχνει μια σύγκριση μεταξύ ενός βιολογικού νευρώνα και ενός τεχνητού νευρώνα, επισημαίνοντας τα δομικά τους στοιχεία.



**Εικόνα 3.14.** Σύγκριση βιολογικού και τεχνητού νευρώνα (Karpathy 2016).

Από το πάνω μέρος της εικόνας 3.14, παρατηρούμε πως ένας βιολογικός νευρώνας αποτελείται από τέσσερις κύριες μονάδες: τους δενδρίτες, το σώμα, τους άξονες και τις συνάψεις. Οι δενδρίτες λαμβάνουν τα σήματα εισόδου και τα μεταφέρουν στο σώμα, όπου επεξεργάζονται. Έπειτα, οι άξονες μεταφέρουν το επεξεργασμένο σήμα σε άλλους νευρώνες μέσω συνάψεων. Οι συνάψεις λειτουργούν ως σύνδεσμοι ανάμεσα στους νευρώνες.

Στον τεχνητό νευρώνα, παρατηρούμε πως οι είσοδοι  $x_1, x_2, \dots, x_n$  παίζουν το ρόλο των δενδριτών, ενώ τα βάρη  $w_1, w_2, \dots, w_n$  αντιστοιχούν στις συνάψεις. Ο υπολογισμός του σταθμισμένου αθροίσματος  $z = \sum_{i=1}^n w_i x_i$  αντιστοιχεί στην επεξεργασία του σήματος στο σώμα του νευρώνα. Τέλος, η έξοδος  $y$  αντιστοιχεί στο σήμα που μεταφέρεται μέσω των αξόνων. Οι έννοιες αυτές αναλύονται λεπτομερώς στη συνέχεια.

## Αρχές Λειτουργίας

### Νευρώνας

Ας ξεκινήσουμε εξετάζοντας τη λειτουργία ενός μεμονωμένου νευρώνα, σε ένα τεχνητό νευρωνικό δίκτυο. Κάθε νευρώνας λαμβάνει εισόδους από άλλους νευρώνες, οι οποίες συμβολίζονται ως  $x_i$ . Κάθε είσοδος πολλαπλασιάζεται με έναν συντελεστή, που ονομάζεται **βάρος** (*weight*) και συμβολίζεται ως  $w_i$ . Αξίζει να σημειωθεί εδώ, πως τα βάρη παίζουν κρίσιμο ρόλο στη λειτουργία του νευρωνικού δικτύου. Συγκεκριμένα, μεγάλα βάρη υποδηλώνουν ότι οι συγκεκριμένες μεταβλητές είναι πιο σημαντικές για την τελική απόφαση του δικτύου. Επομένως, τα βάρη αυτά, πρέπει να προσαρμοστούν κατά την εκπαίδευση του δικτύου, ώστε να αντικατοπτρίζουν την πραγματική σημασία των μεταβλητών.

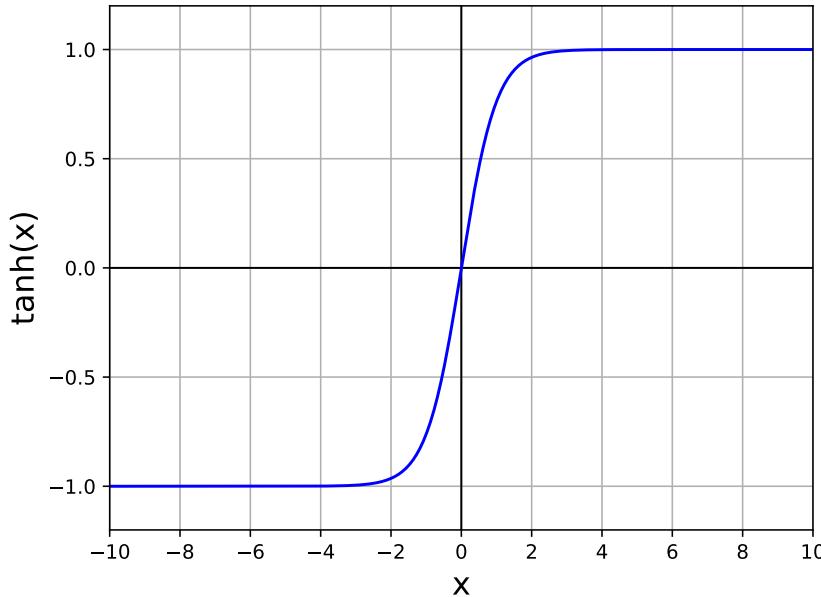
Στη συνέχεια, υπολογίζεται το σταθμισμένο άθροισμα ολων των εισόδων  $z = \sum_{i=1}^n w_i x_i$ . Στο άθροισμα αυτό, προστίθεται κι ένας άλλος παράγοντας, ο οποίος ονομάζεται πόλωση (*bias*) και συμβολίζεται ως  $b$ . Ο παράγοντας αυτός, αποτελεί μία σταθερά και χρησιμοποιείται για την οριζόντια μετατόπιση της εξόδου του νευρώνα. Με αυτόν τον τρόπο, το bias βελτιώνει την προσαρμογή του μοντέλου στα δεδομένα και την ακρίβεια του.

Τέλος, το προηγούμενο αποτέλεσμα περνάει από μια συνάρτηση ενεργοποίησης, για να προκύψει η τελική έξοδος του νευρώνα. Η συνάρτηση ενεργοποίησης (*activation function*) είναι μία μαθηματική συνάρτηση, η οποία συμβολίζεται ως  $f$  και εφαρμόζεται στην έξοδο του νευρώνα, πριν όμως αυτή περάσει στο επόμενο επίπεδο του δικτύου. Η συνάρτηση ενεργοποίησης αποφασίζει αν η έξοδος είναι αρκετά σημαντική, για να περάσει στο επόμενο επίπεδο. Συγκεκριμένα, αν η έξοδος υπερβαίνει ένα καθορισμένο κατώφλι, τότε θα περάσει και λέμε ότι ο νευρώνας «ενεργοποιείται». Οι συναρτήσεις ενεργοποίησης αποτελούν ένα πολύ σημαντικό στοιχείο των νευρωνικών δικτύων, για δύο λόγους. Πρώτον, εισάγουν μη γραμμικότητα (*non linearity*) στο δίκτυο, η οποία είναι εξαιρετικά σημαντική, καθώς χάρη σε αυτήν, καθίσταται δυνατή η επίλυση πολύπλοκων προβλημάτων από τα νευρωνικά δίκτυα. Δεύτερον, ελέγχουν το εύρος των εξόδων των νευρώνων, το οποίο μπορεί να είναι κρίσιμο για τη σταθεροποίηση και την επιτάχυνση της διαδικασίας μάθησης. Υπάρχουν διάφορες συναρτήσεις ενεργοποίησης, με διαφορετικά χαρακτηριστικά η κάθε μία. Η επιλογή της συνάρτησης ενεργοποίησης εξαρτάται από παράγοντες όπως οι απαιτήσεις μάθησης του δικτύου και οι ιδιαιτερότητες του εκάστοτε προβλήματος. Στην πράξη, η επιλογή γίνεται συνήθως έπειτα από δοκιμές και πειραματισμούς, καθώς διαφορετικές συναρτήσεις ενεργοποίησης μπορεί να επιδρούν διαφορετικά στην απόδοση του δικτύου. Δύο από τις πιο διάσημες συναρτήσεις ενεργοποίησης -και οι οποίες χρησιμοποιήθηκαν στο πρόβλημα της αυτόματης στάθμευσης- παρουσιάζονται παρακάτω:

- **Συνάρτηση Tanh:** Η συνάρτηση υπερβολικής εφαπτομένης (*Hyperbolic Tangent - Tanh*) ορίζεται

από την εξίσωση 3.4 και σχεδιάζεται στην Εικόνα 3.15.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \text{ όπου } x \text{ είναι η έξοδος του νευρώνα \quad (3.4)}$$



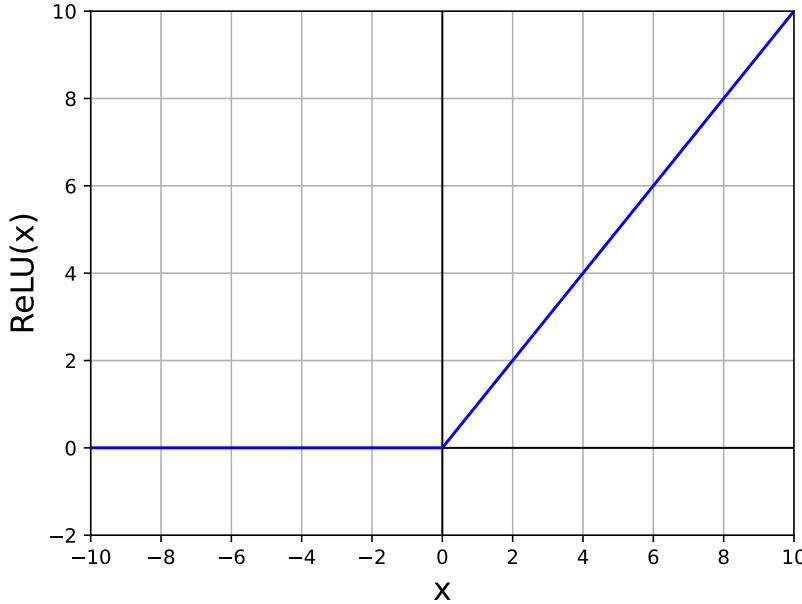
Εικόνα 3.15. Γραφική παράσταση της συνάρτησης Tanh.

Παρατηρούμε πως το εύρος εξόδου της συνάρτησης Tanh είναι  $[-1, 1]$ . Έτσι, η έξοδος της Tanh είναι συμμετρική, με κέντρο το 0. Αυτό συμβάλει στην επιτάχυνση της σύγκλιση του αλγορίθμου κατά την εκπαίδευση. Ακόμα, χάρη σε αυτήν την ίδιότητα της, όταν τα δεδομένα εισόδου είναι κανονικοποιημένα ώστε να έχουν μέση τιμή 0, η εκπαίδευση μπορεί να γίνει πιο αποδοτική. Το βασικό μειονέκτημα της Tanh είναι το πρόβλημα της εξαφάνισης της κλίσης (*vanishing gradients*). Το πρόβλημα αυτό, εμφανίζεται όταν η είσοδος της συνάρτησης γίνεται πολύ μικρή ή πολύ μεγάλη. Τότε, η κλίση της συνάρτησης προσεγγίζει το 0. Αυτό οδηγεί σε πολύ μικρές αλλαγές στα βάρη κατά την εκπαίδευση, κι επομένως η μάθηση του δικτύου γίνεται πολύ αργή ή σταματά.

- **Συνάρτηση ReLU:** Η συνάρτηση Ανορθωμένης Γραμμικής Μονάδας (*Rectified Linear Unit - ReLU*) ορίζεται από την εξίσωση 3.5 και σχεδιάζεται στην Εικόνα 3.16.

$$f(x) = \max(0, x), \text{ όπου } x \text{ είναι η έξοδος του νευρώνα \quad (3.5)}$$

Παρατηρούμε πως το εύρος εξόδου της συνάρτησης ReLU είναι  $[0, \infty)$ . Άρα, δεν υπάρχει ανώτατο όριο για την έξοδο της συνάρτησης κι επομένως, η κανονικοποίηση των δεδομένων εισόδου, είναι ξανά μία καλή πρακτική. Η συνάρτηση ReLU αποτελεί την πιο ευρέως χρησιμοποιούμενη



Εικόνα 3.16. Γραφική παράσταση της συνάρτησης ReLU.

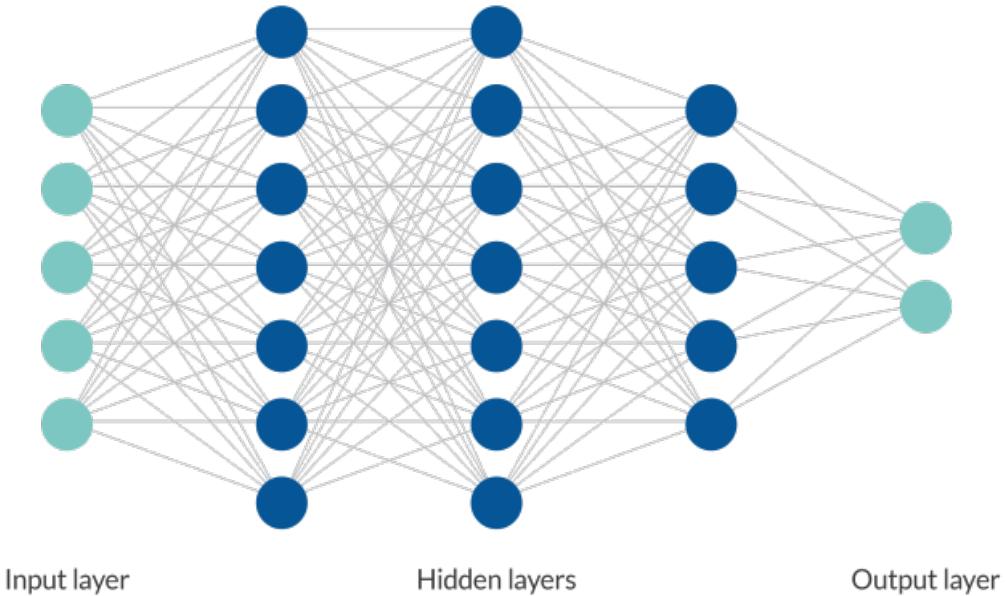
συνάρτηση ενεργοποίησης. Είναι δημοφιλής για τη μικρή απαιτούμενη υπολογιστική ισχύ της, για την επιτάχυνση της σύγκλισης και την αποφυγή του προβλήματος της εξαφάνισης της κλίσης. Ωστόσο, έχει κάποια μειονεκτήματα, με κυριότερο την τάση νεκρών νευρώνων (*dying ReLU problem*), όπου κάποιοι νευρώνες σταματούν να ανταποκρίνονται σε οποιαδήποτε είσοδο και παραμένουν ανενεργοί.

#### Δίκτυο Νευρώνων

Έχοντας μελετήσει τη λειτουργία ενός μεμονωμένου νευρώνα, ας εξετάσουμε τη συνεργασία πολλών μαζί, για τη δημιουργία ενός τεχνητού νευρωνικού δικτύου, όπως αυτό που φαίνεται στην Εικόνα 3.17.

Παρατηρούμε πως τα νευρωνικά δίκτυα αποτελούνται από πολλά επίπεδα (ή στρώματα) νευρώνων, τα οποία συνδέονται μεταξύ τους. Το πρώτο επίπεδο, ονομάζεται επίπεδο εισόδου (*input layer*) και λαμβάνει τα δεδομένα εισόδου του δικτύου. Τα δεδομένα αυτά, στη συνέχεια μεταφέρονται μέσα από μία σειρά επιπέδων που ονομάζονται ενδιάμεσα επίπεδα (*hidden layers*) και είναι υπεύθυνα για την επεξεργασία τους και την εξαγωγή χαρακτηριστικών από αυτά. Τέλος, τα δεδομένα φτάνουν στο τελευταίο επίπεδο, το οποίο ονομάζεται επίπεδο εξόδου (*output layer*) και παράγει την έξοδο του δικτύου.

Ανάλογα με την πολυπλοκότητα του προβλήματος, ένα νευρωνικό δίκτυο μπορεί να αποτελείται από ένα ή περισσότερα ενδιάμεσα επίπεδα. Όταν το δίκτυο αποτελείται από δύο ή περισσότερα ενδιάμεσα επίπεδα, τότε ονομάζεται βαθύ νευρωνικό δίκτυο (*deep neural network*). Αντίστοιχα, το κάθε επίπεδο



**Εικόνα 3.17.** Παράδειγμα τεχνητού νευρωνικού δίκτυου (Comsol 2023).

μπορεί να αποτελείται από λίγους έως και εκατομμύρια νευρώνες.

Ακόμα, ένα νευρωνικό δίκτυο μπορεί να διαφέρει στον τρόπο με τον οποίο συνδέονται οι νευρώνες του. Η πιο διαδεδομένη **τοπολογία**, είναι αυτή των πλήρως συνδεδεμένων επιπέδων (*fully connected layers*). Σε αυτήν, ο κάθε νευρώνας ενός επιπέδου συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου.

### Εκπαίδευση

Τα τεχνητά νευρωνικά δίκτυα μαθαίνουν μέσω της προσαρμογής των βαρών και των πολώσεων τους, κατά τη διάρκεια της εκπαίδευσης. Ας εξετάσουμε τη διαδικασία αυτή βήμα προς βήμα, αναλύοντας παράλληλα ορισμένες σημαντικές έννοιες και παραμέτρους που εμπλέκονται σε αυτήν.

Στο 1<sup>o</sup> βήμα της εκπαίδευσης γίνεται η αρχικοποίηση των βαρών, δηλαδή η ανάθεση τυχαίων τιμών σε αυτά.

Στο 2<sup>o</sup> βήμα της εκπαίδευσης λαμβάνει χώρα η εμπρόσθια διάδοση, κατά την οποία τα δεδομένα εισόδου περνάνε μέσα από το δίκτυο, για να παραχθεί η έξοδος του.

Στο 3<sup>o</sup> βήμα της εκπαίδευσης, υπολογίζεται το σφάλμα της εξόδου, δηλαδή η διαφορά της εξόδου του δίκτυου από την επιθυμητή τιμή της εξόδου<sup>3</sup>. Ο υπολογισμός γίνεται από την **συνάρτηση σφάλματος**

<sup>3</sup>σε προβλήματα επιβλεπόμενης μάθησης, η επιθυμητή τιμή της εξόδου είναι γνωστή για τα δεδομένα εκπαίδευσης. Σε προβλήματα ενισχυτικής μάθησης, ο υπολογισμός του σφάλματος γίνεται με βάση την ανταμοιβή που λαμβάνει το δίκτυο από το περιβάλλον. Περισσότερες λεπτομέρειες στην παράγραφο «Σύνδεση με την Ενισχυτική Μάθηση».

(*loss function*) και στόχος είναι, μέσα από την εκπαίδευση, να ελαχιστοποιηθεί αυτό το σφάλμα. Μερικές από τις πιο διάσημες συναρτήσεις σφάλματος είναι η συνάρτηση Μέσης Τετραγωνικής Απόκλισης (*Mean Squared Error - MSE*), η συνάρτηση Μέσου Απόλυτου Σφάλματος (*Mean Absolute Error - MAE*) και η συνάρτηση σφάλματος Εντροπίας (*Cross-Entropy*) και η επιλογή της εξαρτάται από τον τύπο του προβλήματος.

Στο 4<sup>o</sup> βήμα της εκπαίδευσης λαμβάνει χώρα η οπισθοδρόμικη διάδοση του σφάλματος (*backward propagation* ή *backpropagation*). Κατά τη συγκεκριμένη διαδικασία, υπολογίζεται η κλίση (δηλαδή η μερική παράγωγος) της συνάρτησης σφάλματος ως προς κάθε βάρος του δικτύου. Ο υπολογισμός αυτός γίνεται προς τα πίσω, δηλαδή από το επίπεδο εξόδου προς το επίπεδο εισόδου και αντικατοπτρίζει την επίδραση του κάθε βάρους στο σφάλμα της εξόδου. Αξίζει να σημειωθεί πως, από τεχνικής άποψης, η οπισθοδρόμηση αποτελεί απλώς τη μέθοδο για τον αποτελεσματικό υπολογισμό της κλίσης της συνάρτησης σφάλματος και δεν έχει σχέση με το πως χρησιμοποιείται αυτή στη συνέχεια. Ωστόσο, συχνά, αυτός ο όρος χρησιμοποιείται ελαστικά, για να περιγράψει τη συνολική διαδικασία μάθησης.

Στο 5<sup>o</sup> βήμα της εκπαίδευσης, γίνεται η ενημέρωση των βαρών του δικτύου. Η ενημέρωση των βαρών γίνεται σύμφωνα με τις υπολογισμένες μερικές παραγώγους, προς την κατεύθυνση που μειώνει το σφάλμα και πραγματοποιείται από έναν αλγόριθμο βελτιστοποίησης (*optimization algorithm*). Οι πιο συνηθισμένοι αλγόριθμοι βελτιστοποίησης είναι:

- Ο αλγόριθμος **Κατάβασης Πλαγιάς** (*Gradient Descent*): είναι ο πιο απλός αλγόριθμος βελτιστοποίησης και παρουσιάζει προβλήματα όπως την αργή σύγκλιση και την πιθανότητα να παγιδευτεί σε τοπικά ακρότατα. Αποτέλεσε τη βάση για την ανάπτυξη πιο προηγμένων αλγορίθμων, ενώ υπάρχουν και πολλές παραλλαγές του, όπως ο αλγόριθμος Στοχαστικής Κατάβασης Πλαγιάς (*Stochastic Gradient Descent - SGD*) και ο αλγόριθμος Κατάβασης Πλαγιάς Τεμαχισμένου Ρυθμού (*Mini-batch Gradient Descent*).
- Ο αλγόριθμος **Ορμής** (*Momentum*): αποτελεί επέκταση του SGD, προσθέτοντας έναν όρο «ορμής», δηλαδή ένα τμήμα της προηγούμενης ενημέρωσης στην τρέχουσα ενημέρωση. Βοηθάει στην επιτάχυνση της σύγκλισης και τη μείωση των ταλαντώσεων.
- Ο αλγόριθμος **Διάδοσης Μέσης Τετραγωνικής Απόκλισης** (*Root Mean Square Propagation - RMSprop*): προσαρμόζει τον ρυθμό μάθησης για κάθε παράμετρο ξεχωριστά και κανονικοποιεί το βήμα της ενημέρωσης. Αυτό βοηθάει στην αντιμέτωπιση του προβλήματος της εξαφάνισης της κλίσης (*vanishing gradients*).
- Ο αλγόριθμος **Αδάμ** (*Adam*): συνδυάζει τα πλεονεκτήματα των αλγορίθμων Momentum και RMSprop. Προσαρμόζει το ρυθμό μάθησης για κάθε παράμετρο ξεχωριστά, βασιζόμενος σε προηγούμενες κλίσεις. Είναι πολύ αποδοτικός, καθώς επιτυγχάνει ταχεία σύγκλιση και απαιτεί μικρή ποσότητα μνήμης. Στην πράξη, οι επιδόσεις του Adam συχνά ξεπερνούν τους υπόλοιπους

αλγορίθμους βελτιστοποίησης και για αυτό, αποτελεί μία από τις πιο δημοφιλείς επιλογές σήμερα.

Μία σημαντική παράμετρος των αλγορίθμων βελτιστοποίησης είναι ο ρυθμός μάθησης (*learning rate*), ο οποίος καθορίζει το πόσο γρήγορα τα βάρη του δικτύου προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης. Η προσαρμογή του ρυθμού μάθησης είναι κρίσιμη για να επιτευχθεί ισορροπία μεταξύ της ταχύτητας σύγκλισης και της αποφυγής της υπερπήδησης (*overshooting*) της βέλτιστης λύσης.

Επίσης, αξίζει να σημειωθεί, πως στα πλαίσια αυτής της εργασίας, έγινε η επιλογή του αλγορίθμου βελτιστοποίησης Adam, χάρη στην ικανότητα του να επιτυγχάνει καλά αποτελέσματα, γρήγορα.

Τέλος, τα βήματα 2-5 επαναλαμβάνονται για κάθε τεμάχιο (*batch*) των δεδομένων εκπαίδευσης. Ένα πλήρες πέρασμα όλων των δεδομένων εκπαίδευσης ονομάζεται εποχή (*epoch*). Μπορεί να χρειαστούν πολλές εποχές εκπαίδευσης, ώστε να επιτευχθεί η επιθυμητή απόδοση του δικτύου.

## Υπερ-παράμετροι

Οι παράμετροι του νευρωνικού δίκτυου που ορίζονται πριν την εκπαίδευση, από τον σχεδιαστή του, ονομάζονται υπερ-παράμετροι (*hyperparameters*). Οι παράμετροι αυτοί, έχουν πολύ μεγάλη επίδραση στην τελική αποτελεσματικότητα του δικτύου κι επομένως επιβάλλεται η προσεκτική επιλογή και ρύθμιση τους. Δεν υπάρχουν συγκεκριμένοι κανόνες που να ορίζουν την επιλογή των υπερ-παραμέτρων, κι έτσι αυτή πρέπει να γίνει μέσω δοκιμών και κατάλληλων ρυθμίσεων. Αξίζει να σημειωθεί, πως οι αλλαγές των υπερ-παραμέτρων δεν μπορούν να λάβουν χώρα κατά τη διάρκεια της εκπαίδευσης. Έτσι, αν κριθεί απαραίτητη η αλλαγή κάποιας παραμέτρου του δικτύου, η εκπαίδευση πρέπει να επαναληφθεί από την αρχή. Μερικές από τις βασικότερες υπερ-παραμέτρους ενός νευρωνικού δικτύου είναι:

- **Η αρχιτεκτονική του δικτύου.** Ο όρος «αρχιτεκτονική» ή «δομή» του δικτύου περιλαμβάνει τον αριθμό των επιπέδων, τον αριθμό των νευρώνων σε κάθε επίπεδο καθώς και τη συνδεσμολογία μεταξύ τους. Συνήθως, η αύξηση της πολυπλοκότητας του δικτύου, δηλαδή του πλήθους των επιπέδων και των νευρώνων, βελτιώνει την ικανότητα του να μαθαίνει σύνθετα μοτίβα δεδομένων και αυξάνει έτσι την απόδοση του. Ωστόσο, αυξάνοντας την πολυπλοκότητα του δικτύου, αυξάνεται και ο κίνδυνος της υπερ-προσαρμογής του στα δεδομένα εκπαίδευσης (*overfitting*), με αποτέλεσμα να μην γενικεύει καλά σε νέα δεδομένα (EITCA 2024).
- **Η συνάρτηση ενεργοποίησης:** Η επιλογή της συνάρτησης ενεργοποίησης επηρεάζει την ικανότητα του δικτύου να μάθει και την ταχύτητα της μάθησης.
- **Ο ρυθμός μάθησης:** Η επιλογή ενός υψηλού ρυθμού μάθησης μπορεί να οδηγήσει σε αστάθεια τη διαδικασία εκπαίδευσης, ενώ από την άλλη, ένας χαμηλός ρυθμός μάθησης μπορεί να καθυστερήσει τη σύγκλιση του δικτύου.

### 3.4 Βαθιά Ενισχυτική Μάθηση

---

- **Ο αλγόριθμος βελτιστοποίησης:** Η επιλογή του αλγορίθμου βελτιστοποίησης επηρεάζει την ταχύτητα σύγκλισης και την σταθερότητα της εκπαίδευσης.

Προφανώς, η λίστα αυτή δεν είναι εξαντλητική. Υπάρχουν πολλές ακόμα υπερ-παράμετροι στο εσωτερικό των δικτύων αυτών και κάποιος μπορεί να αναλύσει τη λειτουργία τους σε πολύ μεγαλύτερο βάθος. Ωστόσο, οι παραπάνω παράμετροι αποτελούν αφενός μία καλή αφετηρία για έναν αρχάριο στο αντικείμενο και αφετέρου, θεωρούμε πως με αυτές αξιζει να πειραματιστεί πρώτα κάποιος, στην πράξη.

## Κατηγορίες

Υπάρχουν διαφορετικοί τύποι νευρωνικών δικτύων, οι οποίοι διακρίνονται από την αρχιτεκτονική τους και από τον τύπο προβλημάτων που είναι σχεδιασμένοι να επιλύσουν. Οι πιο κοινοί τύποι νευρωνικών δικτύων σήμερα είναι οι εξής:

- **Τα Νευρωνικά Δίκτυα Πρόσθιας Διάδοσης (*Feedforward Neural Networks*):** Αποτελούν την πιο βασική μορφή νευρωνικών δικτύων. Όπως γίνεται φανερό και από το όνομα τους, χαρακτηριστικό αυτών των δικτύων είναι πως τα δεδομένα εισόδου διαδίδονται μόνο προς την κατεύθυνση της έξοδου, χωρίς να υπάρχουν κύκλοι ή πλάγιες συνδέσεις. Μάλιστα, όταν τα δίκτυα αυτά, αποτελούνται από πολλά επίπεδα, πλήρως συνδεδεμένα μεταξύ τους -δηλαδή στην πλειοψηφία των περιπτώσεων-, τότε ονομάζονται και **Πολυεπίπεδα Νευρωνικά Δίκτυα (*Multilayer Perceptrons - MLPs*)**. Παρά την απλή δομή τους, τα ενδιάμεσα επίπεδα των δικτύων αυτών ενδέχεται να είναι πολύπλοκα κι έτσι, τα μοντέλα αυτά χρησιμοποιούνται σε διάφορες εργασίες, όπως προβλήματα ταξινόμησης και παλινδρόμησης.
- **Τα Συνελικτικά Νευρωνικά Δίκτυα (*Convolutional Neural Networks - CNNs*):** Τα δίκτυα αυτά χρησιμοποιούν την πράξη της συνέλιξης για την επεξεργασία δεδομένων. Είναι ιδιαίτερα αποτελεσματικά στην αναγνώριση προτύπων ή εικόνων και την ανίχνευση αντικειμένων. Έτσι, καθίστανται εξαιρετικά χρήσιμα σε εφαρμογές υπολογιστικής όρασης. Τα CNNs αποτελούνται από τρία βασικά είδη επιπέδων: τα επίπεδα συνέλιξης, υπεύθυνα για την εξαγωγή χαρακτηριστικών από την είσοδο, τα επίπεδα υποδειγματοληψίας, υπεύθυνα για τη μείωση της διάστασης των προηγούμενων χαρακτηριστικών και τα πλήρως συνδεδεμένα επίπεδα, υπεύθυνα για την τελική ταξινόμηση των δεδομένων.
- **Τα Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (*Recurrent Neural Networks - RNNs*):** Τα δίκτυα αυτά διακρίνονται από τους κύκλους ανάδρασης τους, επιτρέποντας έτσι την αποθήκευση πληροφορίας εντός του δικτύου. Με τον τρόπο αυτό, τα RNNs διατηρούν μία μνήμη των προηγούμενων εισόδων και μπορούν να χειριστούν επιτυχώς ακολουθιακά δεδομένα, δηλαδή περιπτώσεις όπου η εισερχόμενη πληροφορία είναι διαδοχικής φύσεως. Για παράδειγμα,

χρησιμοποιούνται ευρέως σε εφαρμογές όπως η αναγνώριση ομιλίας, η πρόβλεψη χρονοσειρών και η επεξεργασία φυσικής γλώσσας.

- Τα **Παραγωγικά Δίκτυα Αντιπαλότητας** (*Generative Adversarial Networks - GANs*): Η βασική διαφοροποίηση των δικτύων αυτών, είναι πως ουσιαστικά, αποτελούνται από δύο ξεχωριστά νευρωνικά δίκτυα. Το πρώτο δίκτυο ονομάζεται «παραγωγός» (*generator*) και είναι υπεύθυνο για τη δημιουργία νέων εικόνων ή κειμένων με βάση ένα σύνολο δεδομένων εκπαίδευσης. Το δεύτερο δίκτυο ονομάζεται «κρίτης» και στόχος του είναι να κρίνει το έργο του παραγωγού, αποφασίζοντας εάν φαίνεται πραγματικό ή ψεύτικο. Η εκπαίδευση ολοκληρώνεται όταν ο κριτής δεν μπορεί να διακρίνει μεταξύ των δεδομένων εκπαίδευσης και των έργων του παραγωγού. Έτσι, τα GANs χρησιμοποιούνται για την παραγωγή νέου περιεχομένου, όπως κείμενο, εικόνες και βίντεο.

Τέλος, αξίζει να σημειώθει πως οι έννοιες που αναλύθηκαν στις προηγούμενες παραγράφους περιγράφουν τη λειτουργία των πολυεπίπεδων νευρωνικών δικτύων (*MLPs*). Οι υπόλοιποι τύποι νευρωνικών δικτύων βασίζονται στις ίδιες αρχές και τρόπο λειτουργίας, αλλά, όπως είδαμε, έχουν και επιπλέον, ειδικές ιδιότητες και μηχανισμούς. Επιλέχθηκε να αναλυθεί πιο συγκεκριμένα η κατηγορία των MLPs, επειδή αποτελούν όχι μόνο τη βάση για την κατανόηση των γενικών αρχών των νευρωνικών δικτύων, αλλά και τον τύπο που χρησιμοποιήθηκε στα πλαίσια αυτής της εργασίας.

### Σύνδεση με την Ενισχυτική Μάθηση

Έχοντας μελετήσει τις θεμελιώδεις αρχές και τον τρόπο λειτουργίας των τεχνητών νευρωνικών δικτύων, ας εξετάσουμε την χρήση τους σε προβλήματα ενισχυτικής μάθησης. Η κεντρική ιδέα είναι πως ο πράκτορας εκφράζεται ως ένα βαθύ νευρωνικό δίκτυο, το οποίο δέχεται ως είσοδο την κατάσταση του περιβάλλοντος και επιστρέφει την επιλεγμένη ενέργεια. Κατά τη διάρκεια της εκπαίδευσης, αναπροσαρμόζονται τα βάρη του νευρωνικού δικτύου, με στόχο την επίτευξη της μέγιστης συνολικής ανταμοιβής.

Ωστόσο, υπάρχουν μερικές διαφοροποιήσεις όσον αφορά την έξοδο του δικτύου και τον τρόπο εκπαίδευσής του, με βάση την κατηγορία του αλγορίθμου ενισχυτικής μάθησης που χρησιμοποιείται. Επομένως, ας εξετάσουμε πιο αναλυτικά τις διαφορές αυτές για τις 3 κατηγορίες model free αλγορίθμων που αναλύσαμε στην υποενότητα 3.3.3.

### Αλγόριθμοι Εκτίμησης Αξίας

Οι πιο χρησιμοποιούμενοι αλγόριθμοι αυτής της κατηγορίας είναι οι αλγόριθμοι Εκτίμησης Αξίας Ζευγών Κατάστασης-Ενέργειας (αλγόριθμοι Q-Learning), όπως για παράδειγμα, ο αλγόριθμος DQN. Οι αλγόριθμοι αυτοί εφαρμόζονται σε περιβάλλοντα με διακριτό χώρο ενεργειών και χρησιμοποιούν ένα δίκτυο αξίας για την εκτίμηση των τιμών  $Q$  των ζευγών κατάστασης-ενέργειας.

- Έξοδος Δικτύου: η έξοδος του νευρωνικού δικτύου αποτελείται από τις τιμές  $Q$  για όλες τις διαθέσιμες ενέργειες στην τρέχουσα κατάσταση του περιβάλλοντος. Η επιλογή ενέργειας γίνεται με βάση κάποια τεχνική όπως η  $\epsilon$ -greedy, για την αντιμέτωπιση του διλήμματος εξερεύνησης-εκμετάλλευσης, όπως έχει αναφερθεί στην υποενότητα 3.3.4.
- Εκπαίδευση Δικτύου: η εκπαίδευση του δίκτυου γίνεται με βάση την ανταμοιβή που λαμβάνει ο πράκτορας από το περιβάλλον. Συγκεκριμένα, η συνάρτηση σφάλματος που χρησιμοποιείται, ονομάζεται Σφάλμα Χρονικών Διαφορών (*Temporal Difference Error - TD Error*) και είναι η διαφορά μεταξύ της εκτιμώμενης τιμής  $Q$  του δικτύου και της πραγματικής τιμής  $Q$ . Η πραγματική τιμή  $Q$  υπολογίζεται από την εξίσωση Bellman ( 3.2), χρησιμοποιώντας την τρέχουσα ανταμοιβή του πράκτορα από το περιβάλλον.

### Αλγόριθμοι Βελτιστοποίησης Πολιτικής

Σε αυτούς τους αλγορίθμους χρησιμοποιείται ένα δίκτυο πολιτικής, δηλαδή το νευρωνικό δίκτυο αντικατοπτρίζει απευθείας την πολιτική του πράκτορα.

- Έξοδος Δικτύου:
  - Διακριτός Χώρος Ενεργειών: η έξοδος του δικτύου είναι μία κατανομή πιθανοτήτων στις διαθέσιμες ενέργειες π.χ.  $P(\text{action}_1) = 0.5$ ,  $P(\text{action}_2) = 0.4$ ,  $P(\text{action}_3) = 0.1$ . Η επιλογή ενέργειας εξαρτάται από τον χαρακτηρισμό της πολιτικής ως ντετερμινιστική ή στοχαστική. Εάν η πολιτική είναι ντετερμινιστική, τότε θα επιλεγεί η ενέργεια με τη μεγαλύτερη πιθανότητα. Αντίθετα, εάν η πολιτική είναι στοχαστική, τότε η επιλογή της ενέργειας θα γίνει μέσω δειγματοληψίας από την κατανομή πιθανοτήτων.
  - Συνεχής χώρος Ενεργειών: η έξοδος του δικτύου, για κάθε ενέργεια, αποτελείται από 2 παραμέτρους μίας Γκαουσιανής κατανομής: τη μέση τιμή της ενέργειας ( $\mu$ ) και την τυπική απόκλιση της ( $\sigma$ ). Για παράδειγμα, αν οι ενέργειες του πράκτορα ήταν η ταχύτητα και η γωνία ενός αυτοκινήτου, τότε η έξοδος του δικτύου θα ήταν της μορφής:  $\mu(\text{velocity})$ ,  $\sigma(\text{velocity})$ ,  $\mu(\text{angle})$ ,  $\sigma(\text{angle})$ . Η επιλογή της συγκεκριμένης τιμής κάθε ενέργειας, εξαρτάται ξανά από την πολιτική του δικτύου. Αν η πολιτική είναι ντετερμινιστική, τότε θα επιλεγεί απλώς η μέση τιμή της ενέργειας. Αντίθετα, αν η πολιτική είναι στοχαστική, τότε η επιλογή της τιμής θα γίνει μέσω δειγματοληψίας από την κατανομή που περιγράφεται από τη μέση τιμή και την τυπική απόκλιση.
- Εκπαίδευση Δικτύου: η εκπαίδευση του δίκτυου γίνεται με βάση την ανταμοιβή που λαμβάνει ο πράκτορας από το περιβάλλον. Συγκεκριμένα, σε κάθε βήμα  $t$  γίνεται μία εκτίμηση για την αθροιστική ανταμοιβή  $G_t$  που θα λάβει ο πράκτορας:

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots \quad (3.6)$$

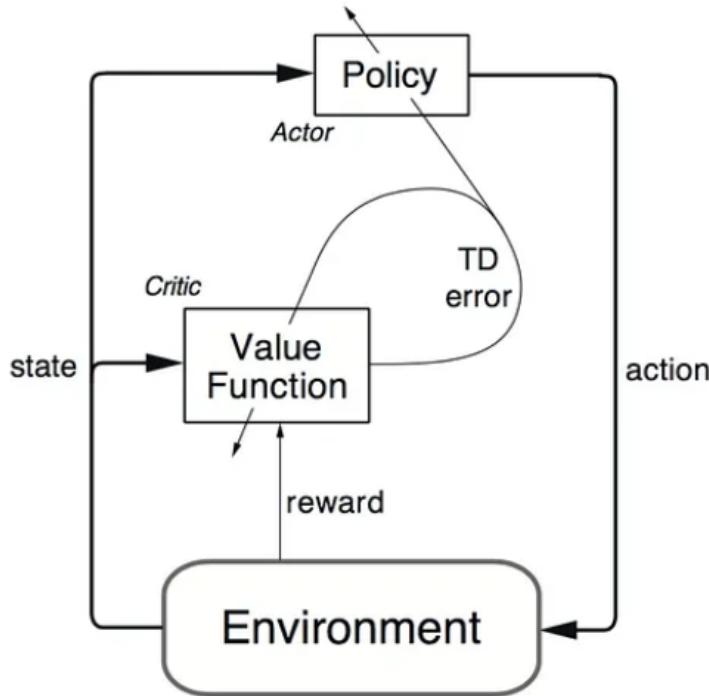
όπου  $R_t$  είναι η τρέχουσα ανταμοιβή και  $\gamma$  ο παράγοντας έκπτωσης. Στόχος είναι η

μεγιστοποίηση της συνάρτησης  $G_t$  κι έτσι, υπολογίζεται η κλίση της ως προς τις παραμέτρους  $\theta$  της πολιτικής. Η κλίση αυτή χρησιμοποιείται για την ενημέρωση των βαρών του δικτύου.

### Αλγόριθμοι Δράστη-Κριτή

Στους αλγορίθμους αυτούς χρησιμοποιούνται δύο νευρωνικά δίκτυα, ο δράστης, υπεύθυνος για την επιλογή της ενέργειας και ο κριτής, υπεύθυνος για την εκτίμηση της αξίας της τρέχουσας κατάστασης ή της επιλεγμένης ενέργειας στην τρέχουσα κατάσταση.

- Έξοδος Δικτύων:
  - Δράστης: η έξοδος του δράστη αποτελεί την πολιτική του πράκτορα. Για την έξοδο του δικτύου και την τελική επιλογή ενέργειας, ισχύουν τα όσα περιγράφησαν νωρίτερα, στους αλγορίθμους βελτιστοποίησης πολιτικής.
  - Κριτής: η έξοδος του κριτή αποτελεί την εκτίμηση μίας αξίας. Σε ορισμένους αλγορίθμους, αυτή η αξία αναφέρεται στην τρέχουσα κατάσταση  $V(s)$ , ενώ σε άλλους αλγορίθμους αναφέρεται στο ζεύγος τρέχουσας κατάστασης-επιλεγμένης ενέργειας  $Q(s, a)$ .
- Εκπαίδευση Δικτύων: η εκπαίδευση γίνεται και σε αυτήν την περίπτωση, με βάση την ανταμοιβή του περιβάλλοντος. Η διαδικασία αποτυπώνεται παραστατικά στην *Εικόνα 3.18*. Συγκεκριμένα, βλέπουμε πως ο κριτής δέχεται ως είσοδο την τρέχουσα κατάσταση και ανταμοιβή του περιβάλλοντος. Έτσι, υπολογίζει το σφάλμα χρονικών διαφορών (*TD Error*), όπως περιγράφηκε νωρίτερα, στους αλγορίθμους εκτίμησης αξίας. Το σφάλμα χρονικών διαφορών χρησιμοποιείται για την ενημέρωση των βαρών του κριτή και του δράστη, όπως φαίνεται από τις καμπύλες που διαπερνούν τα δίκτυα αυτά στην *Εικόνα 3.18*. Με τον τρόπο αυτό, ο κριτής εκπαιδεύει το δίκτυο του, ενώ παρέχει και ανάδραση στον δράστη, η οποία χρησιμοποιείται για την εκπαίδευση του.



**Εικόνα 3.18.** Αλληλεπίδραση δικτύων Δράστη-Κριτή (Sutton και Barto 2018).

Έχοντας πλέον, αναλύσει τα βασικά χαρακτηριστικά της βαθιάς ενισχυτικής μάθησης, καθώς και τον τρόπο λειτουργίας των νευρωνικών δικτύων σε αυτό το πλαίσιο, ας εξετάσουμε τους αλγορίθμους βαθιάς ενισχυτικής μάθησης που χρησιμοποιήθηκαν στην παρούσα εργασία. Συγκεκριμένα, πραγματοποιήθηκαν εκπαίδευσης πρακτόρων με τους εξής αλγόριθμους:

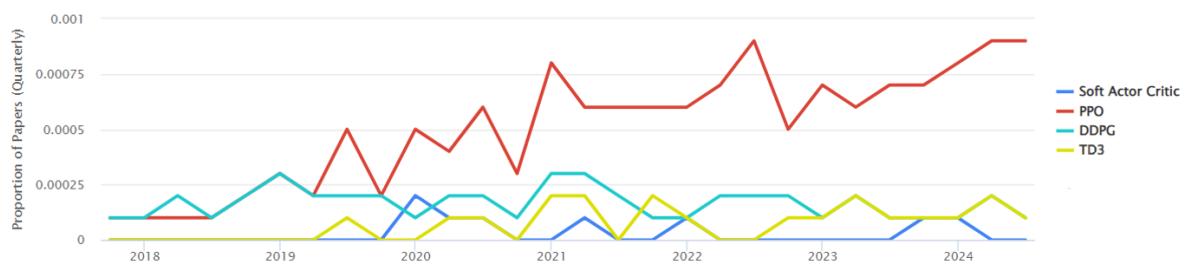
- PPO
- DDPG
- TD3
- SAC

### 3.4.3 Ο αλγόριθμος PPO

Ο αλγόριθμος PPO (*Proximal Policy Optimization*) είναι ένας αποτελεσματικός αλγόριθμος ενισχυτικής μάθησης, που ανήκει στην κατηγορία των αλγορίθμων βελτιστοποίησης πολιτικής. Αναπτύχθηκε το 2017 από την εταιρία OpenAI (Schulman κ.ά. 2017) και ξεχωρίζει για την απλότητά του, τη σταθερότητα του και τις ισχυρές επιδόσεις που πετυχαίνει. Σήμερα, αποτελεί έναν από τους πιο δημοφιλείς αλγορίθμους ενισχυτικής μάθησης. Αυτό αποδεικνύεται και από την [Εικόνα 3.19](#), όπου παρουσιάζεται η συχνότητα των επιστημονικών δημοσιεύσεων των αλγορίθμων ενισχυτικής

μάθησης που χρησιμοποιήθηκαν στην παρούσα εργασία, σε σχέση με τον χρόνο. Παρατηρούμε πως ο αλγόριθμος PPO έχει πολύ μεγαλύτερη συχνότητα χρήσης, σε σχέση με τους υπόλοιπους αλγορίθμους.

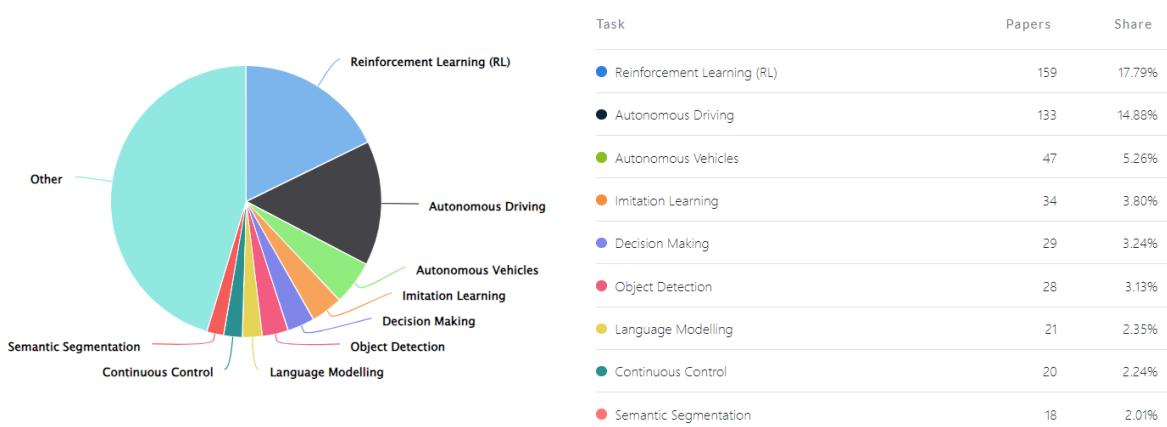
### Usage Over Time



**Εικόνα 3.19.** Συχνότητα χρήσης αλγορίθμων ενισχυτικής μάθησης σε επιστημονικές δημοσιεύσεις (Papers With Code 2024).

Επίσης, στην *Εικόνα 3.20* παρουσιάζονται οι τύποι των προβλημάτων στα οποία χρησιμοποιείται ο αλγόριθμος PPO. Παρατηρούμε πως ο πιο συνήθης τύπος είναι τα προβλήματα ενισχυτικής μάθησης, ακολουθούμενα από τα προβλήματα αυτόνομης οδήγησης και αυτόνομων οχημάτων. Άρα, καταλαβαίνουμε πως ο αλγόριθμος PPO θα είναι κατάλληλος, για την επίλυση του προβλήματος αυτόματης στάθμευσης που αντιμετωπίζουμε στην παρούσα εργασία.

### Tasks



**Εικόνα 3.20.** Τύποι προβλημάτων αλγορίθμου PPO (Papers With Code 2024)<sup>4</sup>.

<sup>4</sup>Αξίζει να σημειωθεί πως τα δύο προηγούμενα γραφήματα προέρχονται από τη σελίδα [paperswithcode](#), η οποία παρέχει χρήσιμες πληροφορίες σχετικά με αλγορίθμους μηχανικής μάθησης, όπως επιστημονικές δημοσιεύσεις, κώδικα, αποτελέσματα κ.α.

### 3.4 Βαθιά Ενισχυτική Μάθηση

---

Ο PPO βελτιστοποιεί την πολιτική του πράκτορα, μέσω της μεγιστοποίησης μιας αντικειμενικής συνάρτησης. Η αντικειμενική συνάρτηση αυτή είναι «περικομμένη» (*clipped*), δηλαδή έχει ένα κατώφλι που περιορίζει την αλλαγή της πολιτικής σε κάθε βήμα. Έτσι, αποφεύγονται μεγάλες, απότομες αλλαγές της πολιτικής και διασφαλίζεται η σταθερότητα της εκπαίδευσης. Η αντικειμενική συνάρτηση του αλγορίθμου PPO παρουσιάζεται στην εξίσωση 3.7, και οι όροι της αναλύονται παρακάτω.

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[ \min \left( r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right] \quad (3.7)$$

- $r_t(\theta)$  αποτελεί το λόγο της νέας πολιτικής προς την παλιά και ισούται με  $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ .
- $\hat{A}_t$  είναι η συνάρτηση πλεονεκτήματος, η οποία εκτιμά το πόσο καλύτερη είναι μια ενέργεια σε μία συγκεκριμένη κατάσταση σε σχέση με τη μέση ενέργεια
- $\text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)$  είναι η συνάρτηση περικοπής, η οποία διασφαλίζει ότι η πολιτική δεν θα υποστεί δραστικές αλλαγές, περιορίζοντας το  $r_t(\theta)$  στο εύρος  $[1 - \epsilon, 1 + \epsilon]$ .
- $\epsilon$  είναι το κατώφλι που περιορίζει την αλλαγή της πολιτικής. Αποτελεί υπερ-παράμετρο του αλγορίθμου και παίρνει μικρές θετικές τιμές (συνήθως 0.2).
- Η συνάρτηση πίν χρησιμοποιείται για την επιλογή του ελάχιστου μεταξύ της μη-περικομμένης και περικομμένης αντικειμενικής συνάρτησης. Έτσι, η τελική αντικειμενική συνάρτηση είναι ένα κάτω φράγμα (δηλαδή μια απαισιόδοξη εκτίμηση) της μη-περικομμένης αντικειμενικής συνάρτησης.

Επίσης, συχνά προστίθεται κι ένας όρος εντροπίας στην αντικειμενική συνάρτηση, ο οποίος ενθαρρύνει τον πράκτορα να εξερευνήσει. Έτσι, διατηρείται μία ποικιλία στις ενέργειες που επιλέγονται από την πολιτική, στα πρώτα στάδια της εκπαίδευσης.

Η μεθοδολογία του αλγορίθμου PPO δίνεται παράκατω υπό μορφή ψευδοκώδικα.

---

#### PPO Algorithm

---

- 1: **Input:** initial policy parameters  $\theta_0$ , clipping threshold  $\epsilon$
- 2: **for**  $k = 0, 1, 2, \dots$  **do**
- 3:     Collect set of partial trajectories  $D_k$  using policy  $\pi_k = \pi(\theta_k)$
- 4:     Estimate advantages  $\hat{A}_t^{\pi_k}$  using any advantage estimation algorithm
- 5:     Compute policy update:  

$$\theta_{k+1} = \arg \max_{\theta} L_{\theta_k}^{\text{CLIP}}(\theta)$$
- 6:     Perform  $K$  steps of minibatch SGD (via Adam), where:

$$L_{\theta_k}^{\text{CLIP}}(\theta) = E_{\tau \sim \pi_k} \left[ \sum_{t=0}^T \min \left( r_t(\theta) \hat{A}_t^{\pi_k}, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t^{\pi_k} \right) \right]$$

- 7: **end for**
- 

Ένα σημαντικό χαρακτηριστικό της εκπαίδευσης του αλγορίθμου PPO, είναι πως δεν εκτελεί μόνο

μία ενημέρωση για κάθε δεδομένο που συλλέγεται. Αντίθετα, πραγματοποιεί πολλαπλές εποχές ενημερώσεων, χρησιμοποιώντας τα ίδια συλλεγμένα δεδομένα. Αυτό καθιστά τον αλγόριθμο πιο αποδοτικό στη χρήση δειγμάτων, σε σύγκριση με άλλες μεθόδους βελτιστοποίησης πολιτικής. Ακόμα, η εκπαίδευση γίνεται on-policy, ενώ χρησιμοποιείται μία στοχαστική πολιτική.

Όσον αφορά την αρχιτεκτονική του νευρωνικού δικτύου, στην αρχική δημοσίευση των (Schulman κ.ά. 2017), χρησιμοποιήθηκε ένα πλήρως συνδεδεμένο MLP με δύο κρυφά επίπεδα των 64 νευρώνων και συνάρτηση ενεργοποίησης Tanh.

Με αυτούς τους τρόπους, ο PPO πετυχαίνει την ανθεκτικότητα και την αξιοπιστία παλαιοτέρων αλγορίθμων βελτιστοποίησης πολιτικής, όπως ο TRPO, αλλά είναι πολύ απλούστερος, ευκολότερος στην υλοποίηση και πετυχαίνει συνολικά, υψηλότερες επιδόσεις. Επιπλέον, είναι αποδοτικός στη χρήση δειγμάτων κι έτσι, μειώνεται το πλήθος των απαιτούμενων αλληλεπιδράσεων με το περιβάλλον. Τέλος, αξιοσημειώτη είναι η ευελιξία του αλγορίθμου, καθώς μπορεί να εφαρμοστεί τόσο σε διακριτούς, όσο και σε συνεχείς χώρους ενεργειών.

#### 3.4.4 Ο αλγόριθμος DDPG

Ο αλγόριθμος DDPG (*Deep Deterministic Policy Gradient*) είναι ένας αλγόριθμος βαθιάς ενισχυτικής μάθησης, που ανήκει στην κατηγορία των αλγορίθμων δράστη-κριτή. Αναπτύχθηκε το 2015 από ερευνητές της εταιρίας DeepMind της Google (Lillicrap κ.ά. 2015). Ο DDPG συνδυάζει στοιχεία των αλγορίθμων βελτιστοποίησης πολιτικής, καθώς και του αλγορίθμου εκτίμησης αξίας DQN. Είναι σχεδιασμένος για περιβάλλοντα με συνεχείς χώρους καταστάσεων και ενεργειών και επομένως, μπορεί να θεωρηθεί ως η εφαρμογή του αλγορίθμου DQN σε συνεχείς χώρους ενεργειών.

Ο DDPG, χρησιμοποιεί ένα δίκτυο δράστη για την επιλογή της ενέργειας. Τα βάρη του δικτύου αυτού συμβολίζονται ως  $\theta^\mu$ . Ακόμα, όπως φανερώνει και το όνομα του αλγορίθμου, ο DDPG χρησιμοποιεί ντετερμινιστική πολιτική. Αυτό σημαίνει, πως από το συνεχές εύρος τιμών της κάθε ενέργειας, ο δράστης δεν επιλέγει μία μέση τιμή  $\mu(s_t)$  και μία τυπική απόκλιση  $\sigma(s_t)$ , αλλά επιλέγει απευθείας μία τιμή  $\mu(s_t)$ . Ωστόσο, ένα σημαντικό χαρακτηριστικό του αλγορίθμου DDPG, είναι πως προσθέτει θόρυβο ( $N_t$ ) στην ενέργεια που επιλέγει ο δράστης, για να ενθαρρύνει την εξερεύνηση του περιβάλλοντος. Ο θόρυβος αυτός συνήθως ακολουθεί κατανομή Gauss ή Ornstein-Uhlenbeck. Επομένως, η ενέργεια που επιλέγει ο δράστης δίνεται από την εξίσωση 3.8:

$$a_t = \mu(s_t | \theta^\mu) + N_t \quad (3.8)$$

Επιπλέον, χρησιμοποιείται ένα δίκτυο κριτή για την εκτίμηση της αξίας των ζευγών κατάστασης-ενέργειας, δηλαδή των τιμών  $Q$ . Τα βάρη του δικτύου αυτού συμβολίζονται ως  $\theta^Q$ .

Ένα ακόμα χαρακτηριστικό του αλγορίθμου είναι η εκπαίδευση off policy. Συγκεκριμένα, αποθηκεύονται προηγούμενες εμπειρίες της μορφής  $(s_t, a_t, R_t, s_{t+1})$  σε μία προσωρινή μνήμη

### 3.4 Βαθιά Ενισχυτική Μάθηση

---

που ονομάζεται *replay buffer*. Κατά την εκπαίδευση, μερικές φορές χρησιμοποιούνται τυχαία δείγματα από τον replay buffer, αντί για τα δεδομένα που συλλέγονται εκείνη τη χρονική στιγμή. Με αυτόν τον τρόπο, περιορίζονται οι συσχετίσεις μεταξύ διαδοχικών δεδομένων και επιτυγχάνεται μεγαλύτερη σταθερότητα στην εκπαίδευση.

Μία άλλη, σημαντική ιδιότητα του αλγορίθμου DDPG είναι η κανονικοποίηση τεμαχίων (*batch normalization*). Αυτή η τεχνική χρησιμοποιείται για την κανονικοποίηση των εισόδων στα δίκτυα δράστη και κριτή, έτσι ώστε να έχουν μονάδική μέση τιμή και διακύμανση. Αυτό βοηθά στην αύξηση της σταθερότητάς της εκπαίδευσης.

Τέλος, αξιοσημείωτη αποτελεί η χρήση δύο δίκτυων στόχου (*target networks*), ένα για τον δράστη και ένα για τον κριτή. Τα δίκτυα αυτά έχουν βάρη  $\theta^{\mu'}$  και  $\theta^{Q'}$  αντίστοιχα, τα οποία ενημερώνονται αργά, με βάση τα βάρη των κύριων δίκτυων. Μάλιστα, χρησιμοποιείται ένας παράγοντας  $\tau$  που ελέγχει τον ρυθμό ενημέρωσης των βαρών των δίκτυων στόχων και παίρνει μικρές τιμές (συνήθως  $\tau = 0.001$ ). Με τον τρόπο αυτό, οι ενημερώσεις γίνονται πιο ομαλές και αποφεύγονται μεγάλες ταλαντώσεις κατά την εκπαίδευση.

Τα δίκτυα στόχου χρησιμοποιούνται στον υπολογισμό της τιμής  $Q$  στόχου ( $y_t$ ), ο οποίος γίνεται μέσα από την εξίσωση 3.9:

$$y_t = R_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1} | \theta^{\mu'}) | \theta^{Q'}) \quad (3.9)$$

Επομένως, το δίκτυο του κριτή ενημερώνεται μέσω της ελαχιστοποίησης του τετραγώνου της διαφοράς, μεταξύ της πρόβλεψης  $Q$  του κριτή και της τιμής στόχου  $y_t$ . Η αντίστοιχη συνάρτηση σφάλματος, παρουσιάζεται στην εξίσωση 3.10:

$$L(\theta^Q) = \frac{1}{N} \sum_t (y_t - Q(s_t, a_t | \theta^Q))^2 \quad (3.10)$$

Αντίστοιχα, το δίκτυο του δράστη ενημερώνεται μέσω της μέγιστοποίησης της αναμενόμενης ανταμοιβής, όπως παρουσιάζεται στην εξίσωση 3.11:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) \Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s | \theta^{\mu}) \Big|_{s=s_i} \quad (3.11)$$

Με βάση αυτά, η μεθοδολογία του αλγορίθμου DDPG δίνεται και στην επόμενη σελίδα, υπό μορφή ψευδοκώδικα.

Συνολικά, η προσέγγιση του αλγορίθμου DDPG είναι απλή, εύκολη στην υλοποίηση και επεκτάσιμη σε δύσκολα προβλήματα, μεγάλης διαστασιμότητας. Ακόμα, οι τεχνικές που χρησιμοποιεί, όπως τα δίκτυα στόχων και ο replay buffer, βελτιώνουν σε μεγάλο βαθμό την ανθεκτικότητα της μάθησης.

Ωστόσο, οι ίδιες τεχνικές προκαλούν και μερικές αδυναμίες του αλγορίθμου, όπως η αργή σύγκλιση και η ανάγκη για μεγάλο αριθμό επαναλήψεων. Επιπλέον, πρόκληση αποτελεί η εξερεύνηση του

περιβάλλοντος, καθώς η ντετερμινιστική πολιτική του αλγορίθμου μπορεί να οδηγήσει σε τοπικά ακρότατα. Ο προστιθέμενος θόρυβος βοηθάει σε αυτό το πρόβλημα, αλλά συχνά δεν αρκεί για να επιτευχθεί η απαιτούμενη εξερεύνηση.

---

### DDPG Algorithm

---

- 1: **Initialize** critic network  $Q(s, a|\theta^Q)$  and actor  $\mu(s|\theta^\mu)$  with weights  $\theta^Q$  and  $\theta^\mu$
- 2: **Initialize** target networks  $Q'$  and  $\mu'$  with weights  $\theta^{Q'} \leftarrow \theta^Q$ ,  $\theta^{\mu'} \leftarrow \theta^\mu$
- 3: **Initialize** replay buffer  $R$
- 4: **for** episode = 1 to  $M$  **do**
- 5:     **Initialize** a random process  $N$  for action exploration
- 6:     **Receive** initial observation state  $s_1$
- 7:     **for**  $t = 1$  to  $T$  **do**
- 8:         **Select** action  $a_t = \mu(s_t|\theta^\mu) + N_t$  according to the current policy and exploration noise
- 9:         **Execute** action  $a_t$  and observe reward  $r_t$  and new state  $s_{t+1}$
- 10:         **Store** transition  $(s_t, a_t, r_t, s_{t+1})$  in  $R$
- 11:         **Sample** a random minibatch of  $N$  transitions  $(s_i, a_i, r_i, s_{i+1})$  from  $R$
- 12:         **Calculate** target Q-value:  $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
- 13:         **Update** critic by minimizing the loss:  $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
- 14:         **Update** the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q) \Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu) \Big|_{s=s_i}$$

- 15:         **Update** the target networks:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$$

- 16:         **end for**
  - 17:     **end for**
- 

**Εικόνα 3.21.** Ψευδοκώδικας του αλγορίθμου DDPG.

#### 3.4.5 Ο αλγόριθμος TD3

Ο αλγόριθμος TD3 (*Twin Delayed Deep Deterministic policy gradient*) αποτελεί μία βελτιωμένη έκδοση του αλγορίθμου DDPG, η οποία αναπτύχθηκε το 2018 από ερευνητές των πανεπιστημίου του Montreal και του Amsterdam (Fujimoto, Hoof, και Meger 2018). Είναι σχεδιασμένος για περιβάλλοντα με συνεχείς χώρους καταστάσεων και ενεργειών, ενώ στόχος του είναι να αντιμετωπίσει μερικά από τα μειονεκτήματα του DDPG, όπως την υπερεκτίμηση της συνάρτησης αξίας και την αστάθεια κατά την εκπαίδευση. Ο TD3 πετυχαίνει αυτόν τον στόχο, μέσω της εισαγωγής τριών βασικών τροποποιήσεων: της εξομάλυνσης της πολιτικής στόχου (*target policy smoothing*), της χρήσης

### 3.4 Βαθιά Ενισχυτική Μάθηση

---

μίας περικομμένης εκδοχής του αλγορίθμου Διπλού Q-learning (*clipped Double Q-learning*) και της καθυστέρησης της ενημέρωσης της πολιτικής (*delayed policy updates*).

Όπως και ο προκάτοχός του, ο TD3 ανήκει στην κατηγορία των αλγορίθμων δράστη-κριτή. Μάλιστα, χρησιμοποιεί ένα δίκτυο δράστη και δύο δίκτυα κριτή -εξού και ο χαρακτηρισμός *Twin* (Δίδυμος) στο όνομα του αλγορίθμου-. Οι παράμετροι του δικτύου δράστη συμβολίζονται ως  $\phi$  και τα βάρη των δικτύων κριτή ως  $\theta_1$  και  $\theta_2$ .

Η πρώτη τροποποίηση του αλγορίθμου είναι η εξομάλυνση της πολιτικής στόχου (*target policy smoothing*). Η τεχνική αυτή εισάγεται στον υπολογισμό της ενέργειας στόχου  $\tilde{a}$ , η οποία θα χρησιμοποιηθεί στη συνέχεια, στον υπολογισμό της τιμής  $Q$  στόχου. Συγκεκριμένα, στον υπολογισμό της  $\tilde{a}$ , προστίθεται ένας μικρός βαθμός θορύβου ( $\epsilon$ ), στην ενέργεια που επέλεξε ο δράστης. Ο θόρυβος περιορίζεται σε ένα εύρος  $[-c, c]$ , όπου  $c$  είναι μία σταθερά, ώστε η ενέργεια στόχου να μην αποκλίνει υπερβολικά από την ενέργεια που επέλεξε ο δράστης. Η διαδικασία αυτή φαίνεται στην εξίσωση 3.12:

$$\tilde{a} \sim \pi_{\phi'}(s') + \epsilon, \epsilon \sim \text{clip}(N(0, \tilde{\sigma}), -c, c) \quad (3.12)$$

Με αυτόν τον τρόπο, μειώνεται η διασπορά στις εκτιμήσεις των τιμών  $Q$  και αποτρέπεται η πολιτική από την εκμετάλλευση πιθανών αιχμών της συνάρτησης  $Q$ , που συνήθως οφείλονται σε σφάλματα υπερεκτίμησης από τα δίκτυα κριτή.

Η δεύτερη τροποποίηση του αλγορίθμου αφορά το πρόβλημα της υπερεκτίμησης των τιμών  $Q$  (*overestimation bias*). Το πρόβλημα αυτό, εμφανίζεται συχνά στους αλγορίθμους εκτίμησης αξίας και περισσότερο, όταν χρησιμοποιούνται συναρτήσεις προσέγγισης, όπως τα νευρωνικά δίκτυα. Το αποτέλεσμα του, είναι η υιοθέτηση υποβέλτιστων πολιτικών από τον πράκτορα. Τέτοια σφάλματα, εμφανίζονται και στην περίπτωση των αλγορίθμων δράστη-κριτή, αφού στους συγκεκριμένους αλγόριθμους, η πολιτική ενημερώνεται με βάση την εκτίμηση των τιμών  $Q$ . Ο αλγόριθμος TD3 επιδιώκει να ελαχιστοποιήσει το πρόβλημα της υπερεκτίμησης, χρησιμοποιώντας μία περικομμένη (*clipped*) εκδοχή του αλγορίθμου *Double Q-learning*. Σύμφωνα με την εκδοχή αυτή, η τιμή  $Q$  στόχου ( $y$ ) υπολογίζεται από την εξίσωση 3.13:

$$y = R + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a}) \quad (3.13)$$

Επομένως, η τιμή  $Q$  στόχου υπολογίζεται ως το ελάχιστο από τις τιμές  $Q$  που επιστρέφουν τα δύο δίκτυα κριτή. Αυτός ο κανόνας, μπορεί να προκαλέσει το αντίθετο φαινόμενο, την υποεκτίμηση των τιμών  $Q$ . Ωστόσο, αυτό είναι σαφώς προτιμότερο από την υπερεκτίμηση, καθώς οι τιμές των υποεκτιμημένων ενεργειών δεν θα μεταδοθούν κατά τη μάθηση, αφού οι ενέργειες με μικρές τιμές  $Q$  αποφεύγονται από την πολιτική.

Με βάση την τιμή  $Q$  στόχου ( $y$ ), τα δίκτυα κριτή ενημερώνουν τις παραμέτρους τους  $\theta_1$  και  $\theta_2$ , μέσω της ελαχιστοποίησης του τετραγώνου της διαφοράς μεταξύ της πρόβλεψης του κριτή και της τιμής

$Q$  στόχου. Η αντίστοιχη συνάρτηση σφάλματος παρουσιάζεται στην εξίσωση 3.14:

$$\theta_i = \arg \min_{\theta_i} \frac{1}{N} \sum (y - Q_{\theta_i}(s, a))^2 \quad (3.14)$$

Η τελευταία τροποποίηση του TD3 είναι η καθυστέρηση της ενημέρωσης της πολιτικής. Συγκεκριμένα, το δίκτυο του δράστη (και κατ' επέκταση το δίκτυο στόχου του δράστη) δεν ενημερώνεται σε κάθε βήμα της εκπαίδευσης, αλλά ανά  $d$  βήματα (προτείνεται  $d = 2$ ). Αυτή η καθυστέρηση βοηθά στη σταθεροποίηση της διαδικασίας μάθησης, επιτρέποντας στα δίκτυα κριτή να συγκλίνουν καλύτερα, πριν γίνει η ενημέρωση της πολιτικής. Έτσι, οι λιγότερο συχνές ενημερώσεις της πολιτικής, θα χρησιμοποιούν μία εκτίμηση της τιμής  $Q$  με μικρότερη διακύμανση και συνεπώς, θα αποτελούν ενημερώσεις υψηλότερης ποιότητας. Οι ενημερώσεις αυτές της πολιτικής, δίνονται στην εξίσωση 3.15:

$$\nabla_{\phi} J(\phi) = \frac{1}{N} \sum \nabla_a Q_{\theta_1}(s, a) \Big|_{a=\pi_{\phi}(s)} \nabla_{\phi} \pi_{\phi}(s) \quad (3.15)$$

Με βάση τη διαδικασία που περιγράφηκε προηγουμένως, η μεθοδολογία του αλγορίθμου TD3 δίνεται παράκατω υπό μορφή ψευδοκώδικα.

---

### TD3 Algorithm

---

- 1: Initialize critic networks  $Q_{\theta_1}$ ,  $Q_{\theta_2}$ , and actor network  $\pi_{\phi}$  with random parameters  $\theta_1$ ,  $\theta_2$ ,  $\phi$
  - 2: Initialize target networks  $\theta'_1 \leftarrow \theta_1$ ,  $\theta'_2 \leftarrow \theta_2$ ,  $\phi' \leftarrow \phi$
  - 3: Initialize replay buffer  $B$
  - 4: **for**  $t = 1$  to  $T$  **do**
  - 5:     Select action with exploration noise  $a \sim \pi_{\phi}(s) + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma)$  and observe reward  $R$  and new state  $s'$
  - 6:     Store transition tuple  $(s, a, R, s')$  in  $B$
  - 7:     Sample mini-batch of  $N$  transitions  $(s, a, R, s')$  from  $B$
  - 8:      $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon$ ,  $\epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  - 9:      $y \leftarrow R + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  - 10:    Update critics  $\theta_i \leftarrow \arg \min_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  - 11:    **if**  $t \bmod d$  **then**
  - 12:       Update  $\phi$  by the deterministic policy gradient:  

$$\nabla_{\phi} J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a) \Big|_{a=\pi_{\phi}(s)} \nabla_{\phi} \pi_{\phi}(s)$$
  - 13:       Update target networks:
  - 14:        $\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$
  - 15:        $\phi' \leftarrow \tau \phi + (1 - \tau) \phi'$
  - 16:    **end if**
  - 17: **end for**
  - 18: **end for**
- 

Συνολικά, οι τροποποιήσεις του αλγορίθμου TD3 είναι απλές και εύκολες στην υλοποίηση, ενώ παίζουν καθοριστικό ρόλο στην αύξηση της απόδοσης σε σχέση με τον αλγόριθμο DDPG. Συγκεκριμένα, οι αλλαγές του TD3 μειώνουν την αστάθεια που μπορεί να εμφανιστεί στον DDPG,

προωθούν την ανθεκτικότητα της εκπαίδευσης και τελικά, οδηγούν σε πιο αξιόπιστες πολιτικές.

Παρόλα αυτά, ο αλγόριθμος είναι πιο πολύπλοκος από τον DDPG, με χαρακτηριστικά παραδείγματα την χρήση δύο δικτύων κριτή και την καθυστέρηση της ενημέρωσης της πολιτικής. Επομένως, το υπολογιστικό κόστος του αλγορίθμου είναι υψηλότερο, ενώ η εκπαίδευση του μπορεί να απαιτεί περισσότερο χρόνο. Ακόμα, οι νέες τροποποιήσεις εισάγουν περισσότερες υπερ-παραμέτρους, οι οποίες πρέπει να ρυθμιστούν με προσοχή, ώστε να μην επηρεάσουν αρνητικά την απόδοση του αλγορίθμου.

### 3.4.6 Ο αλγόριθμος SAC

Ο αλγόριθμος SAC (*Soft Actor-Critic*) είναι ένας αλγόριθμος βαθιάς ενισχυτικής μάθησης, που ανήκει στην κατηγορία των αλγορίθμων δράστη-κριτή. Αναπτύχθηκε το 2018 από ερευνητές του πανεπιστημιού του Berkeley και της Google (Haarnoja κ.ά. 2018). Αποτελεί μία επέκταση των παραδοσιακών μεθόδων δράστη-κριτή, που ενσωματώνει την κανονικοποίηση της εντροπίας στην αντικειμενική συνάρτηση, κάνοντας την πολιτική στοχαστική και ενθαρρύνοντας την εξερεύνηση. Εφαρμόζεται σε περιβάλλοντα με συνεχείς χώρους καταστάσεων και ενεργειών. Ξεχωρίζει για την αποδοτικότητα του στη χρήση δειγμάτων, την ανθεκτικότητα της εκπαίδευσης του και τις επιδόσεις του σε πολύπλοκα προβλήματα, όπως η πλοιήγηση ρομπότ.

Ο SAC βασίζεται στο πλαίσιο της μέγιστης εντροπίας της ενισχυτικής μάθησης. Σε αυτό το πλαίσιο, ο δράστης στοχεύει στη μεγιστοποίηση της αναμενόμενης ανταμοιβής, ενώ ταυτόχρονα μεγιστοποιεί την εντροπία. Με άλλα λόγια, ο δράστης προσπαθεί να επιτύχει στην εργασία του, ενεργώντας όσο το δυνατόν πιο τυχαία. Με τον τρόπο αυτό, ο αλγόριθμος επιχειρεί να πετυχεί μία ισορροπία στο δίλημμα εξερεύνησης-εκμετάλλευσης. Το πλαίσιο αυτό, εμφανίζεται και στην αντικειμενική συνάρτηση του αλγορίθμου SAC, η οποία παρουσιάζεται στην εξίσωση 3.16:

$$J(\pi) = \mathbb{E}_\pi \left[ \sum_t R(s_t, a_t) - \alpha \log(\pi(a_t | s_t)) \right] \quad (3.16)$$

Οι όροι της αντικειμενικής συνάρτησης αναλύονται παρακάτω:

- Ο όρος  $\sum_t R(s_t, a_t)$  αποτελεί την αναμενόμενη αθροιστική ανταμοιβή του περιβάλλοντος. Παρατηρούμε ότι ο πράκτορας στοχεύει στη μεγιστοποίηση της.
- Ο όρος  $-\alpha \log(\pi(a_t | s_t))$  αποτελεί την εντροπία της πολιτικής, την οποία ο πράκτορας επίσης στοχεύει να μεγιστοποιήσει.
- Ο όρος  $\alpha$  είναι η παράμετρος θερμοκρασίας (*temperature parameter*), η οποία καθορίζει την επίδραση του όρου εντροπίας έναντι του όρου της ανταμοιβής και συνεπώς, ελέγχει τη στοχαστικότητα της βέλτιστης πολιτικής. Η κλασική, μέγιστη αναμενόμενη ανταμοιβή, των συμβατικών αλγορίθμων ενισχυτικής μάθησης, ανακτάται στο όριο όταν  $\alpha \rightarrow 0$ .

Προκειμένου να μεγιστοποιήσει την παραπάνω αντικειμενική συνάρτηση, ο αλγόριθμος SAC εκπαιδεύει πέντε διαφορετικά νευρωνικά δίκτυα:

- 1 δίκτυο δράστη, το οποίο αντιπροσωπεύει την πολιτική του πράκτορα  $\pi(a_t|s_t)$ ,
- 2 δίκτυα κριτή, τα οποία εκτιμούν την αξία των ζευγών κατάστασης-ενέργειας, δηλαδή τις τιμές  $Q(s_t, a_t)$  και
- 2 δίκτυα κριτή στόχου (*target networks*), τα οποία αποτελούν καθυστερημένα αντίγραφα των δικτύων κριτή και χρησιμοποιούνται στον υπολογισμό του σφάλματος, κατά την εκπαίδευση των δικτύων κριτή. Τα δίκτυα αυτά συμβάλουν σημαντικά στη σταθεροποίηση της εκπαίδευσης.

Ακόμα, ο αλγόριθμος SAC εκτελείται off-policy, δηλαδή τα δεδομένα που συλλέγονται από την αλληλεπίδραση με το περιβάλλον, αποθηκεύονται σε μία μνήμη (*replay buffer*) και μπορούν να χρησιμοποιηθούν πολλές φορές κατά την εκπαίδευση.

Αναλυτικότερα, η μεθοδολογία που ακολουθεί ο αλγορίθμος SAC δίνεται παράκατω υπό μορφή ψευδοκώδικα. Έχουν προστεθεί σχόλια στον ψευδοκώδικα, προκειμένου να γίνει πιο κατανοητή η λειτουργία του αλγορίθμου.

---

### SAC Algorithm

---

```

1: Input:  $\theta_1, \theta_2, \phi$                                 ▷ Initial critic and actor networks parameters
2:  $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$     ▷ Initialize target network weights
3:  $D \leftarrow \emptyset$                                      ▷ Initialize an empty replay buffer
4: for each iteration do
5:   for each environment step do
6:      $a_t \sim \pi_\phi(a_t|s_t)$                                ▷ Sample action from the policy
7:      $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$                       ▷ Receive transition from the environment
8:      $D \leftarrow D \cup \{(s_t, a_t, R(s_t, a_t), s_{t+1})\}$  ▷ Store the transition in the replay buffer
9:   end for
10:  for each gradient step do
11:     $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$  for  $i \in \{1, 2\}$  ▷ Update critic networks Q-functions
12:     $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$            ▷ Update actor network policy
13:     $\alpha \leftarrow \alpha - \lambda_\alpha \hat{\nabla}_\alpha J(\alpha)$           ▷ Adjust temperature
14:     $\bar{\theta}_i \leftarrow \tau \theta_i + (1 - \tau) \bar{\theta}_i$  for  $i \in \{1, 2\}$  ▷ Update target network weights
15:  end for
16: end for
17: Output:  $\theta_1, \theta_2, \phi$                                 ▷ Optimized critic and actor networks parameters

```

---

Συνολικά, ο SAC διαθέτει πολλά στοιχεία στον τρόπο λειτουργίας του, τα οποία δημιουργούν σημαντικά πλεονεκτήματα σε σχέση με άλλους αλγορίθμους.

Αρχικά, η ιδιότητα του SAC ως off policy αλγορίθμου και η χρήση του replay buffer, τον καθιστά ιδιαίτερα αποδοτικό στη χρήση δειγμάτων και μειώνει τον αριθμό των απαιτούμενων αλληλεπιδράσεων με το περιβάλλον. Επιπλέον, η εκπαίδευση γίνεται με τη χρήση μιας στοχαστικής

### 3.5 Σύνοψη

---

πολιτικής, η οποία ενθαρρύνει τον πράκτορα να εξερευνήσει το περιβάλλον.

Επίσης, όπως αναφέρθηκε και παραπάνω, η χρήση της εντροπίας στην αντικειμενική συνάρτηση, βοηθάει στην αποδοτική εξερεύνηση του περιβάλλοντος από τον πράκτορα και αποτρέπει την πρόωρη σύγκλιση της πολιτικής του. Αυτό είναι κρίσιμο σε περιβάλλοντα με αραιές ανταμοιβές, όπως το πρόβλημα της αυτόματης στάθμευσης.

Ακόμα, η χρήση διαφορετικών νευρωνικών δικτύων για την εκτιμήση των τιμών  $Q$ , συμβάλλει στην αποφυγή της υπερεκτίμησης των τιμών τους, το οποίο αποτελεί συχνό πρόβλημα σε τέτοιου είδους μεθόδους. Έτσι, ο αλγόριθμος επιτυγχάνει πιο σταθερή και αξιόπιστη μάθηση.

Ωστόσο, ο αλγόριθμος SAC παρουσιάζει και ορισμένα μειονέκτημα. Το βασικότερο από αυτά αποτελεί η πολυπλοκότητα του, η οποία οδηγεί σε αυξημένο υπολογιστικό κόστος. Συγκεκριμένα, η ανάγκη ενημέρωσης πολλαπλών νευρωνικών δικτύων για τη συνάρτηση αξίας, επιβαρύνει σημαντικά το υπολογιστικό έργο.

Τέλος, αδυναμία του αλγορίθμου είναι η ευαισθησία στην κλίμακα των ανταμοιβών (*reward scale*). Συγκεκριμένα, για μικρές τιμές των ανταμοιβών, ο πράκτορας αποτυγχάνει να οξιοποιήσει το σήμα ανταμοιβής κι έτσι, υποβαθμίζεται σημαντικά η απόδοση του. Αντίθετα, για μεγάλες τιμές ανταμοιβών, ο πράκτορας μαθαίνει γρήγορα στην αρχή, αλλά συχνά συγκλίνει σε τοπικά ελάχιστα, λόγω έλλειψης επαρκούς εξερεύνησης. Επομένως, είναι απαραίτητη η σωστή κλιμάκωση των ανταμοιβών, ώστε ο πράκτορας να ισορροπήσει την εξερεύνηση και την εκμετάλλευση και να πετύχει τελικά, καλύτερη απόδοση. Παρόλα αυτά, είναι θετικό πως η κλίμακα των ανταμοιβών αποτελεί τη μόνη υπερ-παράμετρο, που απαιτεί στην πράξη, προσεκτική ρύθμιση από τον σχεδιαστή του συστήματος.

## 3.5 Σύνοψη

Στο κεφάλαιο αυτό, δώσαμε το θεωρητικό υπόβαθρο της παρούσας εργασίας. Ξεκινήσαμε ορίζοντας την Τεχνητή Νοημοσύνη και εξετάζοντας τις εφαρμογές και κατηγορίες της. Υστερα, αναλύσαμε το πεδίο της Μηχανικής Μάθησης και μελετήσαμε τους τρεις τύπους της. Έπειτα, επικεντρώθηκαμε στο κύριο αντικείμενο της εργασίας, τον τομέα της Ενισχυτικής Μάθησης. Ορίσαμε τις βασικές έννοιες του πεδίου και εξετάσαμε τη χρήση των νευρωνικών δικτύων σε αυτό. Τέλος, παρουσιάσαμε τις κατηγορίες αλγορίθμων ενισχυτικής μάθησης και αναλύσαμε τους 5 αλγορίθμους που υλοποιήσαμε στην παρούσα εργασία.

Η θεωρία που αναλύθηκε σε αυτό το κεφάλαιο αποτελεί τη βάση για την κατανόηση των εκπαιδεύσεων που διεξήχθησαν και των αποτελεσμάτων τους, τα οποία θα δούμε στα επόμενα κεφάλαια. Επομένως, από το επόμενο κεφάλαιο, αρχίζει το πρακτικό κομμάτι της εργασίας, ξεκινώντας με την παρουσιάση του περιβάλλοντος εκπαίδευσης, δηλαδή του παιχνιδιού που κατασκευάστηκε.

## 4 Το παιχνίδι

Έχοντας αναλύσει τις βασικές αρχές της ενισχυτικής μάθησης, γίνεται σαφές πως απαιτείται ένα περιβάλλον για την εκπαίδευση του πράκτορα. Το περιβάλλον αυτό, θα περιέχει μία εργασία (*task*) που θα αποτελεί στόχο για τον πράκτορα, ενώ αυτός θα λαμβάνει ανταμοιβή ανάλογα με την απόδοσή του. Στην πλειοψηφία των περιπτώσεων εργασιών ενισχυτικής μάθησης, το περιβάλλον εκπαίδευσης είναι ένα παιχνίδι. Στα παιχνίδια, ο στόχος του πράκτορα είναι καλά καθορισμένος και η απόδοσή του μπορεί να μετρηθεί με σαφήνεια. Επιπλέον, τα παιχνίδια παρέχουν ένα περιβάλλον με πολλές διαφορετικές καταστάσεις, κι έτσι μπορεί να αξιολογηθεί η γενίκευση του εκπαιδευμένου πράκτορα. Ακόμα, τα παιχνίδια αυτά, αποτελούν πολλές φορές προσωμοιώσεις πραγματικών προβλημάτων. Έτσι, η εκπαίδευση του πράκτορα σε ένα παιχνίδι, αποτελεί ένα πρώτο βήμα πριν την εφαρμογή του σε καταστάσεις του πραγματικού κόσμου. Στο πνεύμα αυτό, το περιβάλλον εκπαίδευσης που επιλέχθηκε για τη συγκεκριμένη εργασία, το παιχνίδι αυτόματης στάθμευσης, αποτελεί ένα πραγματικό πρόβλημα, του οποίου η λύση θα διευκόλυνε την καθημερινότητα των ανθρώπων.

Στο παρόν κεφάλαιο, παρουσιάζουμε το παιχνίδι στάθμευσης (*Parking Game*), που κατασκευάστηκε στα πλαίσια αυτής της εργασίας, ως περιβάλλον εκπαίδευσης πρακτόρων. Αρχικά, στην Ενότητα 4.1 θα εξηγήσουμε τους λόγους κατασκευής ενός καινούργιου παιχνιδιού. Έπειτα, στην Ενότητα 4.2 θα αναλύσουμε τους κανόνες του παιχνιδιού και στην Ενότητα 4.3 θα σχολιάσουμε κάποιες ενδιαφέρουσες λεπτομέρειες της υλοποίησης του, κάποιες προκλήσεις που προέκυψαν και τον τρόπο με τον οποίο αντιμετωπίστηκαν και τις τελικές αδυναμίες του παιχνιδιού. Σε όλες αυτές τις ενότητες, θα αναλύουμε κάθε απόφαση που πάρθηκε και θα την τεκμηριώνουμε.

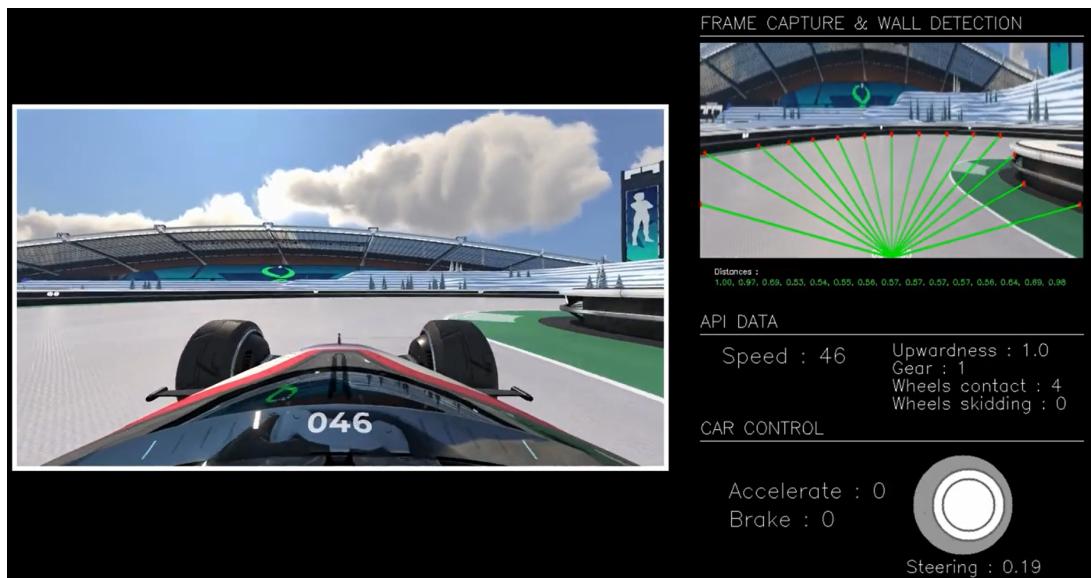
### 4.1 Αιτιολογία κατασκεύης παιχνιδιού

Σε αυτήν την Ενότητα, θα απαντήσουμε στο ερώτημα: «Γιατί να κατασκευάσουμε ένα παιχνίδι από το μηδέν;». Η απάντηση σε αυτό το ερώτημα θα διασαφηνίσει τον τρόπο σκέψης μας και θα εξηγήσει τις επιλογές που κάναμε. Παράλληλα, θα βοηθήσει τους ενδιαφερόμενους αναγνώστες να πάρουν τη δική τους απόφαση, γνωρίζοντας τα δεδομένα σε κάθε περίπτωση.

Μία πρώτη επιλογή είναι να χρησιμοποιήσουμε ένα παιχνίδι που έχει ήδη κατασκευαστεί. Για παράδειγμα, τα βίντεο του (Yosh 2022), σχετικά με την εκπαίδευση πρακτόρων στο δημοφιλές

#### 4.1 Αιτιολογία κατασκεύης παιχνιδιού

παιχνίδι αγώνων ταχύτητας «Trackmania», αποτέλεσε προσωπικά, μία από τις πρώτες επαφές με τον τομέα της ενισχυτικής μάθησης. Ωστόσο, το κυριότερο πρόβλημα που παρουσιάζεται σε αυτήν την περίπτωση, είναι η επικοινωνία με το παιχνίδι. Συγκεκριμένα, ο πράκτορας πρέπει να δέχεται ως είσοδο, την κατάσταση του παιχνιδιού και να επιστρέψει σε αυτό ως έξοδο, την επιλεγμένη ενέργεια του. Κάποιοι αλγόριθμοι ενισχυτικής μάθησης, όπως ο DQN, είναι ικανοί να εκπαιδεύονται απευθείας από εικόνες (*raw pixels input*). Επομένως, θα ήταν απαραίτητη η ανάπτυξη ή χρήση ενός προγράμματος που θα δημιουργούσε ένα στιγμιότυπο της οθόνης του παιχνιδιού (*screen capture*) σε κάθε καρέ (*frame*), θα το μετέτρεπε σε ένα διάνυσμα από pixels και θα το έδινε ως είσοδο στον πράκτορα. Από την άλλη, έτσι η είσοδος του αλγορίθμου είναι της μορφής διανύσματος χαρακτηριστικών του παιχνιδιού (*feature engineering*), τότε μπορεί να χρειαστεί κάποια εφαρμογή μηχανικής όρασης (*computer vision*) για να εξαχθούν αυτά τα δεδομένα. Αντίστοιχα, για την έξοδο του πράκτορα θα μπορούσε να χρησιμοποιηθεί κάποια βιβλιοθήκη όπως η PyAutoGU, για τον έλεγχο του ποντικιού και του πληκτρολογίου. Ακόμα, κάποια εμπορικά παιχνίδια διαθέτουν ειδικές διεπαφές (*APIs*), που επιτρέπουν την αλληλεπίδραση με εξωτερικά προγράμματα, δίνοντας τους πρόσβαση σε δεδομένα του παιχνιδιού (*game state data*). Αυτό είναι σαφώς πολύ χρήσιμο, όμως μπορεί να αποδειχθεί ανεπαρκές. Για παράδειγμα, στην Εικόνα 4.1 φαίνεται ο τρόπος επικοινωνίας με το παιχνίδι Trackmania, του προηγούμενου βίντεο.



Εικόνα 4.1. Επικοινωνία πράκτορα με το παιχνίδι Trackmania (Yosh 2022).

Στο αριστερό μέρος της Εικόνας 4.1 διακρίνεται η εικόνα του παιχνιδιού, όπως την βλέπουμε οι άνθρωποι. Στο δεξιό μέρος, φαίνεται η επικοινωνία με το API του παιχνιδιού, για την είσοδο κι έξοδο του πράκτορα. Ωστόσο, όπως φαίνεται στο πάνω-δεξιά μέρος της εικόνας, κρίθηκε απαραίτητη η δημιουργία ενός προγράμματος καταγραφής στιγμιοτύπων της οθόνης και μέτρησης της απόστασης του αυτοκινήτου από τους τοίχους της πίστας μέσα από αυτά. Επομένως, τα δεδομένα του API ήταν

ανεπαρκή και για αυτό το λόγο, αναπτύχθηκε ένα πρόγραμμα μηχανικής όρασης.

Με βάση τα παραπάνω, παρατηρούμε πως η χρήση ενός ήδη υπάρχοντος παιχνιδιού, παρουσιάζει προκλήσεις και μπορεί να απαιτήσει προϋπάρχουσες γνώσεις σε τομείς, όπως η μηχανική όραση και η επεξεργασία εικόνας. Επίσης, δυσκολίες μπορεί να προκύψουν και σε μεταγενέστερα στάδια της εκπαίδευσης και οι όποιες αλλαγές πρέπει να γίνουν, θα είναι δύσκολα εφαρμόσιμες.

Για όλους τους ανωτέρω λόγους, θεωρούμε πως είναι προτιμότερο να χρησιμοποιηθεί ένα περιβάλλον ειδικό για την εκπαίδευση πρακτόρων, το οποίο θα έχει προβλέψει τις ανάγκες που υπάρχουν, και θα παρέχει τα απαραίτητα εργαλεία. Η πιο διαδεδομένη πλατφόρμα αυτού του είδους, είναι το Gymnasium της OpenAI (Brockman κ.ά. 2016).

Η πλατφόρμα Gymnasium της OpenAI αποτελεί μία βιβλιοθήκη γραμμένη σε γλώσσα Python και είναι ένα σημαντικό εργαλείο για την έρευνα στον τομέα της ενισχυτικής μάθησης. Χρησιμοποιείται για την ανάπτυξη και σύγκριση αλγορίθμων ενισχυτικής μάθησης, παρέχοντας μια μεγάλη ποικιλία περιβαλλόντων, από απλές εργασίες, όπως παιχνίδια της πλατφόρμας Atari, εως σύνθετα προβλήματα ελέγχου, όπως προσομοιώσεις ρομποτικών εφαρμογών. Η βιβλιοθήκη παρέχει μία απλή διεπαφή, με μία σειρά από υλοποιημένες συναρτήσεις, που επιτρέπουν την εύκολη εκπαίδευση και αξιολόγηση πρακτόρων ενισχυτικής μάθησης. Η βιβλιοθήκη είναι σήμερα εξαιρετικά δημοφιλής και χρησιμοποιείται σχεδόν σε κάθε εργασία ή άρθρο που αφορά την ενισχυτική μάθηση. Επομένως, μία απλή και αξιόπιστη λύση, είναι η χρήση της βιβλιοθήκης OpenAI Gymnasium για την παροχή του περιβάλλοντος και όλων των απαραίτητων μεθόδων, για την αλληλεπίδραση του πράκτορα με αυτό.

Ακόμα, η βιβλιοθήκη OpenAI Gymnasium, δίνει τη δυνατότητα δημιουργίας καινούργιων περιβαλλόντων εκπαίδευσης (*custom gymnasium environment*). Αυτό σημαίνει, πως μπορεί κάποιος να κατασκευάσει σε κώδικα ένα δικό του παιχνίδι και στη συνέχεια, να το μετατρέψει σε περιβάλλον συμβατό με τη βιβλιοθήκη. Η μετατροπή δεν περιλαμβάνει μεγάλες διαφορές, παρά μόνο την υλοποίηση ορισμένων κλάσεων και μεθόδων. Με τον τρόπο αυτό, θα είναι δυνατή η αξιοποίηση των πολύ χρήσιμων εργαλείων της βιβλιοθήκης, σε ένα δικό μας παιχνίδι. Αυτή ήταν και η επιλογή που προτιμήθηκε σε αυτήν την εργασία.

Συγκεκριμένα, κατασκευάστηκε ένα παιχνίδι σε γλώσσα Python, με χρήση της βιβλιοθήκης Pygame. Στη συνέχεια, ο κώδικας προσαρμόστηκε κατάλληλα, ώστε να μετατραπεί το παιχνίδι σε καινούργιο περιβάλλον εκπαίδευσης. Αυτή η διαδικασία παρέχει το βασικό πλεονέκτημα της εξοικείωσης με τον κώδικα του παιχνιδιού και της ελευθερίας αλλαγής του. Συγκεκριμένα, δεν υπάρχει κανένας περιορισμός, στο πως θα εκφράζονται οι καταστάσεις του περιβάλλοντος ή ποιες θα είναι οι δυνατές ενέργειες του πράκτορα. Ακόμα, μπορεί το παιχνίδι να προσαρμοστεί στις ανάγκες της εκπαίδευσης, προσθέτοντας, αφαιρώντας ή τροποποιώντας χαρακτηριστικά του. Επομένως, η κατασκευή ενός παιχνιδιού από το μηδέν, παρέχει μία ευελιξία και ελευθερία κινήσεων και επιλογών, η οποία δεν προσφέρεται σε καμία άλλη περίπτωση. Ένα άλλο πλεονέκτημα, είναι πως η γλώσσα Python αποτελεί την πιο ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού στον τομέα της

#### 4.1 Αιτιολογία κατασκεύης παιχνιδιού

---

μηχανικής μάθησης. Συνεπώς, πέρα από τη συμβατότητα με τη βιβλιοθήκη OpenAI Gymnasium, ένα παιχνίδι γραμμένο σε Python παρέχει εύκολη συμβατότητα με δημοφιλείς βιβλιοθήκες αλγορίθμων μηχανικής μάθησης, όπως η PyTorch και το TensorFlow. Τέλος, είναι καλό να αναλογιστεί κανείς το υπολογιστικό κόστος της εκτέλεσης του παιχνιδιού. Συγκεκριμένα, οι αλγόριθμοι βαθιάς ενισχυτικής μάθησης είναι ιδιαίτερα κοστοβόροι και απαιτούν σημαντική επεξεργασία, σε πόρους όπως ο επεξεργαστής (*CPU*) και η κάρτα γραφικών (*GPU*). Επομένως, είναι σημαντικό το παιχνίδι να είναι απλό και να μην προσθέτει μεγάλη επιβάρυνση στο υπολογιστικό σύστημα, όταν εκτελείται. Αυτό είναι ένα ακόμα πλεονέκτημα της χρήσης της βιβλιοθήκης Pygamer, η οποία είναι ελαφριά και αποδοτική.

Για όλους τους παραπάνω λόγους, κρίθηκε προτιμότερη η κατασκευή ενός παιχνιδιού σε Python και εκ των υστέρων, θα τη συνιστούσα ως την καλύτερη επιλογή για την εκπαίδευση πρακτόρων ενισχυτικής μάθησης. Παρόλα αυτά, αξίζει να αναφερθούμε σε δύο ενδιαφέρουσες, εναλλακτικές λύσεις.

Η πρώτη εναλλακτική επιλογή αφορά τη χρήση της μηχανής παιχνιδιών (*game engine*) Unity. Η Unity είναι μία ισχυρή μηχανή, που επιτρέπει την κατασκευή παιχνιδιών σε 2 ή 3 διαστάσεις, μέσω της χρήσης της γραφικής διεπαφής της, καθώς και της συγγραφής αρχείων κώδικα (*scripts*) σε γλώσσα C#. Η βιβλιοθήκη Unity Machine Learning Agents είναι ένα έργο ανοιχτού κώδικα, που επιτρέπει τη δημιουργία περιβαλλόντων εκπαίδευσης για ευφυείς πράκτορες μηχανικής μάθησης. Μάλιστα, παρέχει έτοιμες μεθόδους, όπως υλοποιημένους αλγορίθμους ενισχυτικής μάθησης, μέσω μίας διεπαφής σε γλώσσα Python. Επομένως, η κατασκευή ενός παιχνιδιού σε Unity και η εκπαίδευση μέσω της βιβλιοθήκης ML Agents, αποτελεί μία καλή εναλλακτική λύση και σίγουρα, το τελικό αποτέλεσμα θα είναι πιο εντυπωσιακό, όσον αφορά την αισθητική του. Ωστόσο, η Unity είναι μία πολύπλοκη μηχανή και η εξοικείωση με αυτήν, καθώς και η κατασκευή ενός παιχνιδιού μέσω αυτής, απαιτεί σημαντικό χρόνο. Δεδομένου ότι η εκπαίδευση πρακτόρων ενισχυτικής μάθησης είναι ήδη μία δύσκολη και χρονοβόρα διαδικασία, είναι προτιμότερο να αφιερωθεί ικανός χρόνος σε αυτήν, παρά στη δημιουργία ενός πιο ρεαλιστικού ή πιο όμορφου παιχνιδιού. Επομένως, ο στόχος της εργασίας είναι τέτοιος, ώστε η επιλογή αυτή να ήταν προτιμότερη, μόνο αν ήμασταν ήδη εξοικειωμένοι με το περιβάλλον της Unity και τη γλώσσα C#.

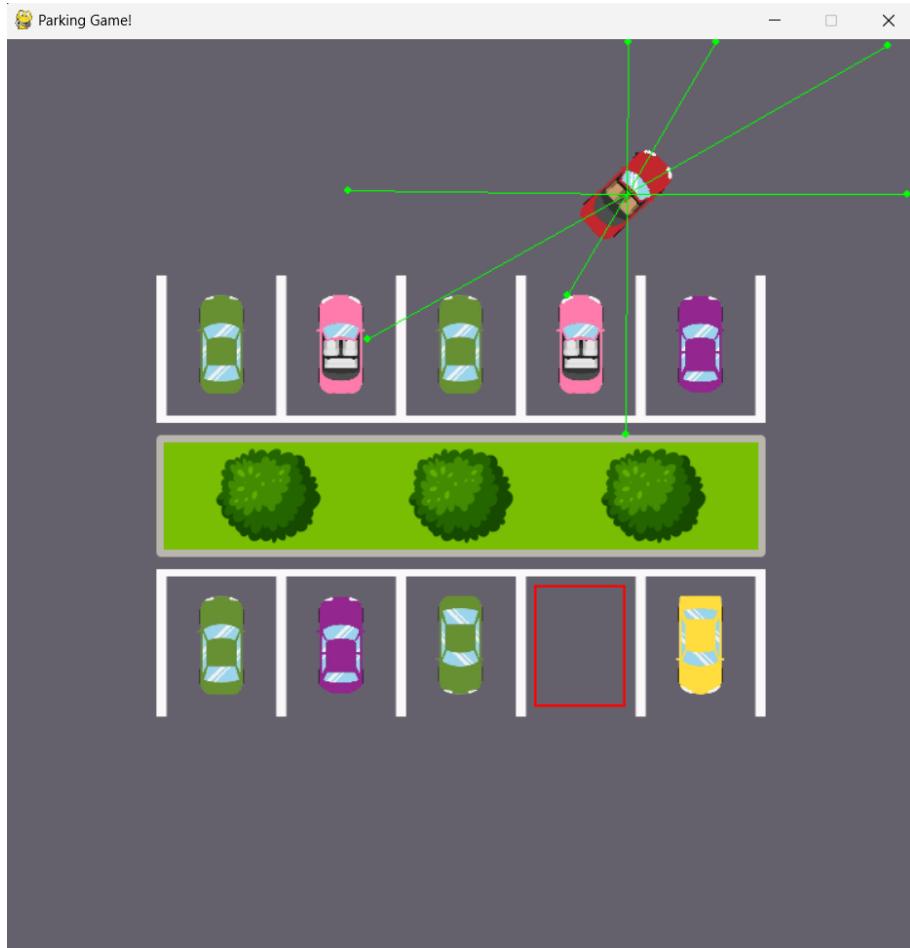
Η δεύτερη εναλλακτική λύση, είναι η χρήση ενός προγράμματος προσομοιωτή. Για την περίπτωση της αυτόνομης οδήγησης που εξετάζεται σε αυτήν την εργασία, δημοφιλή τέτοια περιβάλλοντα αποτελούν το TORCS (*The Open Racing Car Simulator*) και το Carla (*Car Learning to Act*). Υπάρχει πληθώρα δημοσιεύσεων μηχανικής μάθησης, που χρησιμοποιούν αυτούς τους προσωμοιωτές. Η χρήση τους παρέχει επιπλέον ρεαλισμό, με μηχανισμούς φυσικής για την ακριβή αναπαράσταση της κίνησης των οχημάτων και προσφέρει ποικιλία δυνατοτήτων, όπως η χρήση πραγματικών αισθητήτων διαφόρων ειδών (π.χ. κάμερες, LIDAR, GPS). Ακόμα, διατίθενται διαφορετικά σενάρια εκπαίδευσης με πολύπλοκες παραμέτρους του πραγματικού κόσμου, όπως φανάρια, διαβάσεις πεζών, κίνηση άλλων οχημάτων και δυσμενείς καιρικές συνθήκες. Επομένως, τα προγράμματα αυτά, αποτελούν την ιδανική επιλογή για εκπαίδευσης με μεγάλη πιστότητα και ακρίβεια

στην αναπαράσταση του πραγματικού κόσμου. Ωστόσο, η χρήση τους απαιτεί σημαντικούς υπολογιστικούς πόρους, όπως εξελιγμένες κάρτες γραφικών και μεγάλα ποσά μνήμης. Ακόμα, όπως και στην περίπτωση της Unity, η εξοικείωση με τα προγράμματα αυτά απαιτεί χρόνο και η δημιουργία ενός περιβάλλοντος εκπαίδευσης θα παρουσιάσει σημαντικές προκλήσεις σε έναν αρχάριο.

## 4.2 Κανόνες παιχνιδιού

### 4.2.1 Βασικοί κανόνες

Σε αυτήν την Ενότητα θα παρουσιάσουμε το παιχνίδι που κατασκευάστηκε και θα εξηγήσουμε τους κανόνες του. Ένα στιγμιότυπο της οθόνης του παιχνιδιού, φαίνεται στην Εικόνα 4.2.



**Εικόνα 4.2.** Στιγμιότυπο του παιχνιδιού «Parking Game».

Παρατηρούμε πως ο χάρτης του παιχνιδιού αποτελεί έναν χώρο στάθμευσης (*parking lot*). Ο χώρος αυτός, περιέχει 10 θέσεις στάθμευσης (*parking spots*), οι οποίες διακρίνονται από τις άσπρες γραμμές

## 4.2 Κανόνες παιχνιδιού

---

στο έδαφος του. Σε κάθε επεισόδιο του παιχνιδιού, μία από αυτές τις θέσεις θα είναι ελεύθερη και τα όρια της θα καθορίζονται από ένα κόκκινο παραλληλόγραμμο. Ο πράκτορας ελέγχει το κόκκινο αυτοκίνητο και ξεκινάει κάθε φορά από τυχαία θέση στον χάρτη και γωνία προσανατολισμού. Στόχος του, είναι να παρκάρει το αυτοκίνητο στην ελεύθερη θέση στάθμευσης. Τότε, ο πράκτορας κερδίζει και ξεκινάει καινούργιο επεισόδιο. Αντίθετα, αν περάσουν 30 δευτερόλεπτα χωρίς να καταφέρει ο πράκτορας να παρκάρει το αυτοκίνητο, τότε χάνει και ξεκινάει καινούργιο επεισόδιο.

Αξίζει να διευκρινίσουμε τι ορίζεται ως επιτυχής στάθμευση στο παιχνίδι. Ο πράκτορας θεωρείται ότι έχει παρκάρει επιτυχώς, όταν το αυτοκίνητο του βρίσκεται μέσα στη θέση στάθμευσης (δηλαδή εντός του αντίστοιχου κόκκινου παραλληλογράμμου) και παραμένει ακίνητο για το χρονικό διάστημα των 2 δευτερολέπτων. Αυτές οι προϋποθέσεις είναι πολύ πιο αυστηρές από αυτές που χρησιμοποιούνται στα περισσότερα περιβάλλοντα αυτόματης στάθμευσης. Για παράδειγμα, στην εργασία των (Mujumdar, Shah, και Cui 2020), θεωρείται επιτυχής στάθμευση, αμέσως μόλις ο πράκτορας αγγίζει το παραλληλόγραμμο της θέσης στάθμευσης. Αντίστοιχα, στην εργασία του (Lai 2018), αρκεί ο πράκτορας να εισέλθει στο παραλληλόγραμμο της θέσης στάθμευσης, χωρίς να απαιτείται κάποιο χρονικό διάστημα ακινησίας. Ωστόσο, οι συνθήκες που επιλέχθηκαν, αντικατοπτρίζουν πιο πιστά την πραγματική διαδικασία στάθμευσης και αποτελούν μεγαλύτερη πρόκληση για τον πράκτορα.

Ακόμα, το παιχνίδι υποστηρίζει τις συγκρούσεις μεταξύ του πράκτορα και των υπολοίπων στοιχείων του χάρτη. Τα στοιχεία αυτά μπορεί να είναι:

- τα υπόλοιπα, σταθμευμένα αυτοκίνητα
- ο κήπος που βρίσκεται στο κέντρο του χάρτη
- η περίμετρος (τα όρια) του χάρτη

Η περίπτωση της σύγκρουσης γίνεται κατανοητή από τους ανθρώπινους παίκτες, καθώς απεικόνιζεται προσωρινά με κόκκινο χρώμα η περιοχή του αντικειμένου με το οποίο συγκρούστηκε το αυτοκίνητο, ενώ ακούγεται κι ένας σχετικός ήχος, με ένταση ανάλογη της ταχύτητας της σύγκρουσης. Τότε, ο πράκτορας θα κινηθεί στιγμιαία προς την αντίθετη κατεύθυνση, με την ταχύτητα της σύγκρουσης του κι ύστερα θα ακινητοποιηθεί. Έπειτα, είναι ελεύθερος να μετακινηθεί ξανά. Επομένως, το επεισόδιο δεν σταματάει στην πρώτη σύγκρουση του πράκτορα, όπως γίνεται στις περισσότερες εργασίες αγώνων ταχύτητας (π.χ. στο (AI-Forge 2022) ή στο (Code-Bullet 2019)). Αντίθετα, το παιχνίδι είναι πιο ανεκτικό στις συγκρούσεις, καθώς στην πράξη, αυτή η στρατηγική αποδείχθηκε πιο αποτελεσματική (βλ. υποενότητα 5.9.1).

Το αυτοκίνητο ελέγχεται με τα βέλη του πληκτρολογίου (UP, DOWN, LEFT, RIGHT), τα οποία χειρίζεται είτε ο άνθρωπος, είτε ο πράκτορας, ανάλογα με την έκδοση του παιχνιδιού που χρησιμοποιείται. Συνολικά, ο τρόπος κίνησης του αυτοκινήτου προσωμοιώνει -σε βαθμό ικανοποιητικό, για τα δεδομένα ενός παιχνιδιού- την πραγματική κίνηση ενός οχήματος.

Επιπλέον, στην Εικόνα 4.2, διακρίνονται επίσης οι ακτίνες (*raycasts*) του αυτοκινήτου, οι οποίες

λειτουργούν ως αισθητήρες (*sensors*) για τον πράκτορα, μετρώντας τις αποστάσεις του από τα υπόλοιπα αντικείμενα στον χάρτη.

Τέλος, με βάση τη λειτουργία που περιγράψαμε, καθώς και τα όσα αναλύθηκαν στην υποενότητα 3.3.2, το περιβάλλον του παιχνιδιού μπορεί να χαρακτηριστεί ως πλήρως παρατηρήσιμο, αιτιοκρατικό, συνεχές, μονοπρακτορικό, επεισοδιακό και με αραιές ανταμοιβές.

### 4.2.2 Γενίκευση

Μία άλλη σημαντική έννοια που πρέπει να αναλογιστούμε, είναι η γενίκευση (*generalization*) του πράκτορα, δηλαδή η ικανότητα του να παρκάρει σε κάθε περίπτωση, ανεξαρτήτων των συνθηκών. Για τη γενίκευση στο παίχνιδι αυτό, παίζουν ρόλο δύο παράγοντες:

1. η αρχική θέση (και γωνία) του πράκτορα και
2. η επιλογή της ελεύθερης θέσης στάθμευσης

Συγκεκριμένα, μέσα από εμπειρικές παρατηρήσεις, καταλήξαμε στα παρακάτω συμπεράσματα για τους διαφορετικούς συνδυασμούς των δύο αυτών παραμέτρων κατά την εκπαίδευση:

1. Όταν είναι καθορισμένες (*fixed*) η αρχική θέση του πράκτορα και η θέση στάθμευσης, τότε ο πράκτορας δεν μαθαίνει να παρκάρει, αλλά αποστηθίζει μία συγκεκριμένη ακολουθία ενεργειών. Με άλλα λόγια, ο πράκτορας απλώς μαθαίνει μία δεδομένη «χορογραφία», (π.χ. UP, UP, LEFT, LEFT, DOWN, RIGHT) και εκτελεί αυτήν την αλληλουχία ενεργειών κάθε φορά. Έτσι, όταν μετά την εκπαίδευση, ο πράκτορας κληθεί να παρκάρει υπό διαφορετικές παραμέτρους, δεν είναι σε θέση να το κάνει, καθώς έχει υπερπροσαρμοστεί (*overfitting*) στις συγκεκριμένες συνθήκες της εκπαίδευσης του.
2. Όταν είναι καθορισμένη η αρχική θέση του πράκτορα, αλλά τυχαία η θέση στάθμευσης, συμβαίνει το ίδιο. Η μόνη διαφορά είναι πως ο πράκτορας θα αποστηθίσει 10 διαφορετικές ακολουθίες ενεργειών και θα εκτελεί την κάθε μία, ανάλογα με την αρχική κατάσταση του.
3. Όταν είναι τυχαία η αρχική θέση του πράκτορα, αλλά καθορισμένη η θέση στάθμευσης, τότε ο πράκτορας πετυχαίνει ένα μερικό βαθμό γενίκευσης. Συγκεκριμένα, όταν μετά την εκπαίδευση, κληθεί να παρκάρει σε μία διαφορετική θέση στάθμευσης, τότε είναι σε θέση να το κάνει, αλλά με μικρότερο ποσοστό επιτυχίας. Αυτό οφείλεται στο ότι οι νέες θέσεις στάθμευσης, ενδέχεται να φέρουν τον πράκτορα σε καταστάσεις με τιμές πολύ διαφορετικές από αυτές που είχε εκπαιδευτεί και τότε, ο πράκτορας δεν καταφέρνει να παρκάρει. Αντίθετα, όταν οι καταστάσεις είναι παρόμοιες με αυτές της εκπαίδευσης του, καταφέρνει να παρκάρει, ακόμα και αν δεν είχε συναντήσει ποτέ, ακριβώς την ίδια κατάσταση στην εκπαίδευση του.
4. Όταν είναι τυχαίες η αρχική θέση του πράκτορα και η θέση στάθμευσης, τότε ο πράκτορας μαθαίνει πραγματικά να παρκάρει κάτω από οποιεσδήποτε συνθήκες.

## 4.2 Κανόνες παιχνιδιού

---

Επομένως, το μεγαλύτερο επίπεδο γενίκευσης επιτυγχάνεται, όταν ο πράκτορας ξεκινάει από τυχαία θέση και η επιλογή της θέσης στάθμευσης γίνεται τυχαία. Για αυτό και επιλέχθηκαν αυτές οι συνθήκες, στους βασικούς κανόνες του παιχνιδιού, όπως είδαμε προηγουμένως.

### 4.2.3 Επίπεδα δυσκολίας

Στην πράξη, αποδείχθηκε πως οι κανόνες του παιχνιδιού, όπως τους περιγράψαμε νωρίτερα, ήταν πολύ δύσκολοι για τους πράκτορες, για να πετύχουν ικανοποιητικά αποτελέσματα. Μία πρακτική λύση που προτείνεται σε τέτοιες περιπτώσεις, είναι η χρήση επιπέδων δυσκολίας. Αυτή η τακτική, διευκολύνει τη σωστή ρύθμιση του συστήματος από τον σχεδιαστή του, καθώς και επιτρέπει στον πράκτορα να εκπαιδευτεί πρώτα σε πιο εύκολες συνθήκες και στη συνέχεια να προχωρήσει σε πιο δύσκολες (*Curriculum Learning*). Θα αναλύσουμε περαιτέρω αυτές τις τεχνικές στην Ενότητα 5.6, όπου θα εξετάσουμε ορισμένες πρακτικές συμβουλες για την καλύτερη εκπαίδευση των πρακτόρων. Προς το παρόν, ας δούμε τα διαφορετικά επίπεδα δυσκολίας που χρησιμοποιήθηκαν στο παιχνίδι μας, και τις τροποποιήσεις του καθενός στους βασικούς κανόνες του παιχνιδιού:

- **Επίπεδο δυσκολίας 1 - Σταθερή θέση στάθμευσης, σταθερή αρχική θέση πράκτορα, άμεση στάθμευση:** πρόκειται για την 1<sup>η</sup> περίπτωση που είδαμε προηγούμενως, με την πρόσθετη διευκόλυνση του άμεσου παρκαρίσματος, δηλαδή τη θεώρηση ότι ο πράκτορας έχει παρκάρει επιτυχώς, μόλις εισέλθει στο παραλληλόγραμμο της θέσης στάθμευσης, χωρίς να απαιτείται κάποιο χρονικό διάστημα ακινησίας. Το επίπεδο αυτό, χρησιμοποιήθηκε απλά ως μία απόδειξη της σωστής λειτουργίας του αλγορίθμου (*proof of concept*), δηλαδή προκειμένου να μας επιβεβαιώσει, ότι ο αλγόριθμος υλοποιήθηκε σωστά και έτσι, ο πράκτορας είναι σε θέση να μάθει χρήσιμες συμπεριφορές.
- **Επίπεδο δυσκολίας 2 - Ημι-τυχαία θέση στάθμευσης, τυχαία αρχική θέση πράκτορα, άμεση στάθμευση:** σε αυτό το επίπεδο δυσκολίας η επιλογή της ελεύθερης θέσης στάθμευσης χαρακτηρίζεται ως ημι-τυχαία. Αυτό σημαίνει πως ακολουθεί την επιλογή της αρχικής θέσης του πράκτορα. Συγκεκριμένα, όταν ο πράκτορας ξεκινάει από το πάνω τμήμα του χάρτη του παιχνιδιού, τότε επιλέγεται ως ελεύθερη θέση μία από τις 5 πάνω θέσεις στάθμευσης. Αντίστοιχα, όταν ο πράκτορας ξεκινάει από το κάτω τμήμα του χάρτη, τότε επιλέγεται ως ελεύθερη θέση μία από τις 5 κάτω θέσεις στάθμευσης. Το επίπεδο αυτό, εξετάζει την ικανότητα γενίκευσης του πράκτορα, σε μία ευκολότερη εκδοχή του παιχνιδιού, όπου ο στόχος του βρίσκεται πιο κοντά στην αρχική του θέση.
- **Επίπεδο δυσκολίας 3 - Τυχαία θέση στάθμευσης, τυχαία αρχική θέση πράκτορα, άμεση στάθμευση:** σε αυτό το επίπεδο δυσκολίας, η επιλογή της ελεύθερης θέσης στάθμευσης, καθώς και η αρχική θέση του πράκτορα είναι τυχαίες. Επομένως, εξετάζεται πλήρως η ικανότητα γενίκευσης του πράκτορα, ανεξάρτητα από την αρχική του απόσταση από την ελεύθερη θέση στάθμευσης. Ωστόσο, η ικανότητα που μαθαίνει ο πράκτορας σε αυτό το επίπεδο δυσκολίας, δεν μπορεί να

χαρακτηριστεί ακόμα ως στάθμευση, καθώς δεν παίζει ρόλο η ταχύτητα του αυτοκινήτου, παρά μόνο η θέση και ο προσανατολισμός του. Επομένως, πρόκειται περισσότερο για μία περίπτωση μετακίνησης από το σημείο A στο σημείο B, όπου τα σημεία A και B είναι τυχαία, με την επιπλέον δυσκολία πως ο προσανατολισμός του αυτοκινήτου πρέπει να είναι συγκεκριμένος στο σημείο B (ώστε να εισέλθει το αυτοκίνητο εντός του παραλληλογράμμου).

- **Επίπεδο δυσκολίας 4** - *Τυχαία θέση στάθμευσης, τυχαία αρχική θέση πράκτορα, κανονική στάθμευση:* σε αυτό το επίπεδο δυσκολίας, πέραν της πλήρους γενίκευσης του πράκτορα, εφαρμόζεται και ο πραγματικός ορισμός της στάθμευσης, εισάγωντας την έννοια της ακινησίας του αυτοκινήτου, για το τυπικό, χρονικό διάστημα των 2 δευτερολέπτων.

Στην πράξη, χρησιμοποιήθηκαν κι άλλες παραλλαγές των παραπάνω επιπέδων δυσκολίας, όμως αυτά αποδείχθηκαν τα πιο χρήσιμα, για την επιτυχή εκπαίδευση των πρακτόρων.

## 4.3 Κατασκευή παιχνιδιού

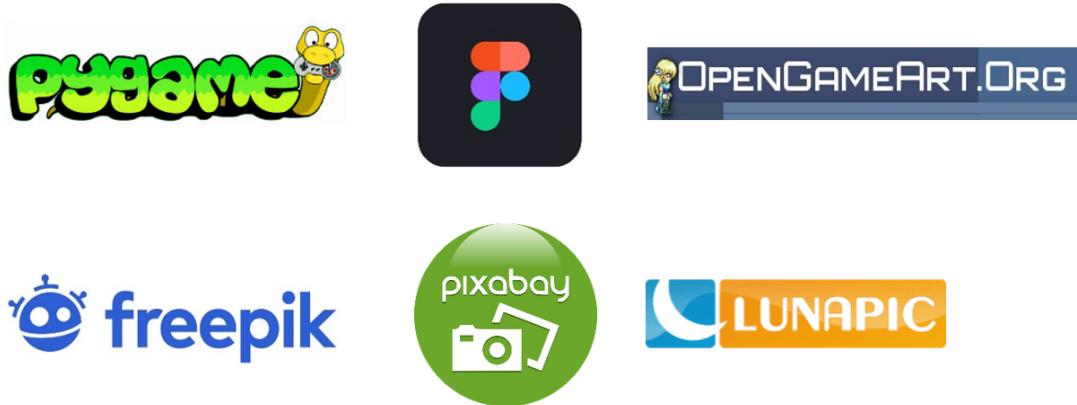
Στη συγκεκριμένη ενότητα, θα δώσουμε περισσότερες λεπτομέρειες σχετικά με τον τρόπο κατασκευής του παιχνιδιού. Αυτό δεν σημαίνει πως θα αναλύσουμε τον κώδικα του παιχνιδιού γραμμή προς γραμμή. Άλλωστε, ο κώδικας περιέχει τα απαραίτητα σχόλια, που καθιστούν κατανοητή τη λειτουργία του. Αντίθετα, θα εξετάσουμε πιο εποπτικά, κάποιες ενδιαφέρουσες αποφάσεις που πάρθηκαν, κατά την κατασκευή του παιχνιδιού και τον τρόπο αντιμετώπισης των προκλήσεων που προέκυψαν. Επομένως, η ενότητα αυτή δεν σχετίζεται άμεσα με την τεχνητή νοημοσύνη, αλλά αποτελεί μία τεκμηρίωση των επιλογών αυτής της εργασίας και σίγουρα, θα φανεί χρήσιμη στους αναγνώστες που ενδιαφέρονται για την κατασκευή παιχνιδιού.

### 4.3.1 Εργαλεία

Τα εργαλεία που χρησιμοποιήθηκαν για την κατασκευή του παιχνιδιού παρουσιάζονται στην *Εικόνα 4.3*.

Η [Pygame](#) αποτελεί τη δημοφιλέστερη βιβλιοθήκη της Python για την κατασκευή παιχνιδιών. Ο λόγος που επιλέχθηκε έναντι άλλων επιλογών, όπως η Pyglet, και η Arcade, είναι η ύπαρξη πληθώρας διαθέσιμου υλικού και παραδειγμάτων (*tutorials*) στο διαδίκτυο. Επίσης, η εξοικείωση με τη βιβλιοθήκη είναι εύκολη και γρήγορη. Για να δώσουμε μία αναφορά, χρειάστηκε περίπου μία εβδομάδα πλήρους ενασχόλησης για την εκμάθηση της Pygame σε ικανοποιητικό επίπεδο, ενώ χρειάστηκαν περί τις δύο εβδομάδες για τη συνολική κατασκευή του παιχνιδιού.

Το εργαλείο σχεδιασμού [Figma](#) χρησιμοποιήθηκε για τη δημιουργία του χάρτη του παιχνιδιού. Η χρήση του Figma, μας έδωσε τη δυνατότητα να δημιουργήσουμε έναν χάρτη εξατομικευμένο στις δικές μας ανάγκες και στόχους. Βέβαια, το μεγαλύτερο πλεονέκτημα του Figma, είναι πως επέτρεψε



Εικόνα 4.3. Εργαλεία κατασκευής παιχνιδιού.

την ακρίβεια σε επιπέδο pixel, στις θέσεις των αντικειμένων και την ανίχνευση συγκρούσεων. Για παράδειγμα, οι θέσεις των αντικειμένων στον κώδικα του παιχνιδιού ορίστηκαν στις αντίστοιχες συντεταγμένες τους, από τον χάρτη του Figma.

Η ιστοσελίδα [OpenGameArt.org](#) παρέχει στοιχεία παιχνιδιών (*game assets*), ελεύθερα από πνευματικά δικαιώματα. Αξιοποιήθηκε για την απόκτηση της μουσικής που παίζει στο παρασκήνιο, καθώς και ηχητικών εφέ, όπως τον ήχο εκκίνησης της μηχανής του αυτοκινήτου, τον ήχο σύγκρουσης και τον ήχο εισόδου του αυτοκινήτου στη θέση στάθμευσης.

Τα τρια τελευταία εργαλεία χρησιμοποιήθηκαν για την εύρεση και επεξεργασία εικόνων, ελεύθερων από πνευματικά δικαιώματα. Συγκεκριμένα, οι εικόνες των αυτοκινήτων του παιχνιδιού προέκυψαν από τις ιστοσελίδες [Freepik](#) και [Pixabay](#), ενώ η επεξεργασία των χρωμάτων τους έγινε μέσω της ιστοσελίδας [Lunapic](#).

#### 4.3.2 Προσωμοίωση φυσικής

Προφανώς, κύριος στόχος αυτής της εργασίας αποτέλεσε η εκπαίδευση και η σύγκριση πρακτόρων ενισχυτικής μάθησης και όχι η ανάπτυξη ενός ρεαλιστικού παιχνιδιού οδήγησης. Παρόλα αυτά, έγινε μία προσπάθεια μίμησης της φυσικής κίνησης του αυτοκινήτου, στο μεγαλύτερο δυνατό βαθμό. Ορισμένα τέτοια παραδείγματα παρουσιάζονται παρακάτω.

#### Περιστροφή

Μία πρώτη πρόκληση αποτέλεσε η περιστροφή του αυτοκινητού. Αρχικά, στην Pygame, οι συντεταγμένες ορίζονται με βάση το πάνω αριστερά σημείο ενός σχήματος. Επομένως, οι υλοποιημένες μέθοδοι που διαθέτει για την περιστροφή των αντικειμένων, περιέστρεφαν το

αυτοκίνητο γύρω από το πάνω αριστερά άκρο του. Χωρίς να μπούμε σε παραπάνω τεχνικές λεπτομέρειες, η συνάρτηση μας `rotate_center` της κλάσης `AbstractCar`, πραγματοποιεί την περιστροφή του αυτοκινήτου γύρω από το κέντρο του. Επίσης, προσαρμόσαμε την γωνία περιστροφής του αυτοκινήτου, ώστε να είναι ανάλογη της ταχύτητας του και ενός παράγοντα περιστροφής. Ο παράγοντας αυτός ορίστηκε μεγαλύτερος σε μικρότερες ταχύτητες του αυτοκινήτου. Ως αποτέλεσμα, το αυτοκίνητο περιστρέφεται πιο εύκολα, όταν κινείται με μικρή ταχύτητα και γίνεται λιγότερο ευέλικτο, όταν κινείται με μεγάλη ταχύτητα. Τέλος, μία μετρική που χρησιμοποιήθηκε για να αξιολογηθεί η περιστροφή του αυτοκινήτου, είναι ο κύκλος στροφής του (*turning circle*). Ο κύκλος στροφής ενός οχήματος, ορίζεται ως ο μικρότερος κύκλος, εντός του οποίου μπορεί να στραφεί πλήρως το όχημα. Αποτελεί μέτρο της ευκολίας, με την οποία το αυτοκίνητο μπορεί να παρκάρει ή να εκτελέσει μία στροφή  $180^\circ$  (*U-turn*). Συνήθως, εκφράζεται από την εξωτερική διάμετρο του, της οποίας η μέση τιμή για τα επιβατικά αυτοκίνητα σήμερα, κυμαίνεται στα  $10.4 - 10.7m$ , σύμφωνα με το (Doss 2024). Στη συνέχεια, εκτελέσαμε ένα πείραμα για να μετρήσουμε τον κύκλο στροφής του αυτοκινήτου στο παιχνίδι μας. Πρώτα από όλα, πρέπει να ορίσουμε την αντιστοίχιση μεταξύ των `pixel` του παιχνιδιού και των μέτρων του πραγματικού κόσμου. Γνωρίζουμε πως στο παιχνίδι, το αυτοκίνητο του πράκτορα έχει διαστάσεις  $82 \times 40$  pixels, ενώ το μέσο μήκος των αυτοκινήτων στον πραγματικό κόσμο είναι  $4.9m$  (Craft 2023). Επομένως, προκύπτει πως 1 pixel του παιχνιδιού αντιστοιχεί σε  $0.0597m$ . Στο πείραμα που διεξάγαμε, εκτελέσαμε το μικρότερο δυνατό ημικύκλιο στο παιχνίδι και μετρήσαμε τις συντεταγμένες του κέντρου του αυτοκινήτου στην αρχική θέση ( $0^\circ$ ) και στην τελική θέση ( $180^\circ$ ). Οι συντεταγμένες αυτές είναι  $(x_1, y_1) = (529, 692)$  και  $(x_2, y_2) = (664, 692)$  αντίστοιχα. Επομένως, η διάμετρος του κύκλου στροφής προκύπτει από την εξίσωση 4.1.

$$d = x_2 - x_1 + 2 \times \text{distance\_to\_wheels} = 664 - 529 + 2 \times 20 = 135 + 40 = 175 \text{ pixels} = 10.457m \quad (4.1)$$

Αξίζει να προσέξουμε πως ο όρος  $2 \times \text{distance\_to\_wheels}$  προστέθηκε, καθώς μετράμε την εξωτερική διάμετρο, κι έτσι πρέπει να συνυπολογίσουμε την απόσταση μεταξύ του κέντρου του αυτοκινήτου και των τροχών του. Επομένως, ο κύκλος στροφής του αυτοκινήτου στο παιχνίδι μας αντιστοιχεί σε  $10.457m$ , το οποίο βρίσκεται σε απόλυτη εναρμόνιση με τα πραγματικά αυτοκίνητα.

## Ταχύτητα και τριβή

Κατά παρόμοιο τρόπο, η μέγιστη ταχύτητα του αυτοκινήτου τέθηκε στα  $6px/frame$ , όπου κάνοντας τις κατάλληλες αντικαταστάσεις, προκύπτει ίση με  $25.79km/h$ . Δεδομένου ότι το αυτοκίνητο κινείται ενός ενός χώρου στάθμευσης, αυτή η μέγιστη ταχύτητα θεωρείται λογική. Μάλιστα, όπως είναι αναμενόμενο, η μέγιστη ταχύτητα του πράκτορα είναι μικρότερη, όταν αυτός κινείται με την όπισθεν. Έτσι, η μέγιστη αρνητική ταχύτητα του πράκτορα ορίστηκε ίση με  $4px/frame$  ή  $17.19km/h$ . Ακόμα, ορίστηκε ένας συντελεστής τριβής, ο οποίος χρησιμοποιείται στη μείωση της ταχύτητας του αυτοκινήτου όταν δεν πατείται κάποιο πλήκτρο. Ο συντελεστής αυτός, τέθηκε ίσος με το μισό του

συντελεστή επιτάχυνσης, ο οποίος χρησιμοποιείται στην αύξηση της ταχύτητας του αυτοκινήτου κατά την πίεση των πλήκτρων.

#### Μηχανισμός σύγκρουσης

Τέλος, στο πνέυμα της ρεαλιστικότητας του παιχνιδιού, έγινε προσπάθεια αποτύπωσης της διαδικασίας των συγκρούσεων του πραγματικού κόσμου. Έτσι, η μέθοδος `bounce` της κλάσης `AbstractCar` εκτελεί έναν μηχανισμό μικρής αναπήδησης του αυτοκινήτου, όταν αυτό συγκρούεται με κάποιο αντικείμενο. Συγκεκριμένα, σύγκρουση ανιχνεύεται στο παιχνίδι όταν, κατά την κίνηση του αυτοκινήτου του πράκτορα, αυτό εισέλθει εντός άλλου αντικειμένου. Τότε, προτού ανανεωθεί η οθόνη, πρέπει να μετακινηθεί το αυτοκίνητο εκτός του αντικειμένου. Η μέθοδος `bounce` επιτυγχάνει αυτόν τον σκοπό, αντιστρέφοντας την ταχύτητα του αυτοκινήτου και μετακινώντας το με αυτήν επανειλημμένα, μέχρι να μην συγκρούεται πλέον με το αντικείμενο. Επείτα, η ταχύτητα του αυτοκινήτου μηδενίζεται. Συνολικά, ο μηχανισμός της σύγκρουσης παρουσίασε αρκετές προκλήσεις και εμφανίστηκαν διάφορες αστοχίες (*bugs*), μέχρι να φτάσει στην τελική του μορφή. Μία από αυτές τις αστοχίες εμφανίζεται ακόμα σε μία ειδική περίπτωση (*corner case*) και αναλύεται στην υποενότητα 4.3.4. Ωστόσο, το αντίκτυπο αυτής της αστοχίας είναι μικρό, ενώ συμβαίνει σπάνια. Επομένως, θεωρούμε πως ο παραπάνω μηχανισμός είναι αρκετά αξιόπιστος και επαρκής για τις ανάγκες του παιχνιδιού.

#### 4.3.3 Ενδιαφέρουσες υλοποιήσεις

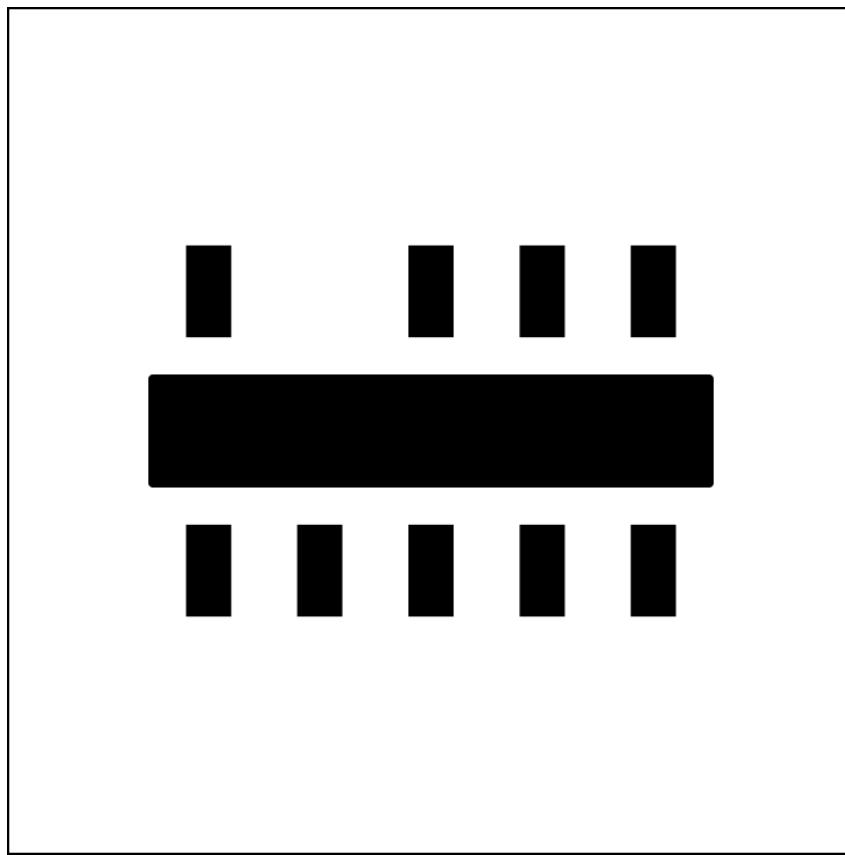
##### Frame rate

Αρχικά, επιλέχθηκε το παιχνίδι να λειτουργεί στα 20 frames per second (fps). Η απόφαση αυτή πάρθηκε, ώστε η εκτέλεση του παιχνιδιού να μην επιβαρύνει σημαντικά τους υπολογιστικούς πόρους του υπολογιστή. Η συχνότητα των 20 fps ήταν η μικρότερη δυνατή, ώστε να εξακολουθεί να φαίνεται ομαλή με το μάτι, η κίνηση του αυτοκινήτου.

##### Ανίχνευση σύγκρουσης

Ενδιαφέρον παρουσιάζει ο μηχανισμός που υλοποιήθηκε για την ανίχνευση σύγκρουσης. Συγκεκριμένα, χρησιμοποιείται η τεχνική της ανίχνευσης σύγκρουσης σε επίπεδο pixel (*pixel-perfect collision detection*), η οποία παρέχει ακριβή ανίχνευση, μέσω της σύγκρισης των pixel των αντικειμένων του παιχνιδιού (π.χ. των εικόνων των αυτοκινήτων). Συγκεκριμένα, ελέγχεται αν υπάρχουν κάποια αδιαφανή pixel δύο αντικειμένων που να συμπίπτουν. Μάλιστα, χρησιμοποιούνται μάσκες των αντικειμένων (*masks*), για τη βελτιστοποίηση και την επιτάχυνση αυτής της διαδικασίας. Οι μάσκες αποτελούν μία δυαδική αναπαράστη μίας εικόνας, όπου κάθε pixel παίρνει την τιμή

Ο εάν είναι διαφανές και την τιμή 1 όταν δεν είναι, δηλαδή όταν βρίσκεται το αντικείμενο στη συγκεκριμένη θέση. Έτσι, η ανίχνευση σύγκρουσης γίνεται με τη σύγκριση των μάσκων των αντικειμένων, ελέγχοντας απλώς αν δύο μάσκες έχουν την τιμή 1 στην ίδια θέση. Επομένως, στο παιχνίδι μας, δημιουργήθηκε μία μάσκα του αυτοκινήτου του πράκτορα, η οποία ανανεώνεται συνεχώς κατά την κίνηση του, και μία στατική μάσκα του χάρτη, η οποία περιέχει τα αντικείμενα με τα οποία μπορεί να προκύψει σύγκρουση. Η στατική αυτή μάσκα θα είναι διαφορετική σε κάθε επεισόδιο, ανάλογα με το ποία θέση στάθμευσης είναι ελεύθερη. Για να δημιουργηθούν αυτές οι στατικές μάσκες, χρησιμοποιήθηκαν εικόνες του χάρτη, μόνο με τα αντικείμενα σύγκρουσης του. Για παράδειγμα, για την περίπτωση όπου η δεύτερη θέση στάθμευσης είναι ελεύθερη, χρησιμοποιήθηκε η *Εικόνα 4.4*.

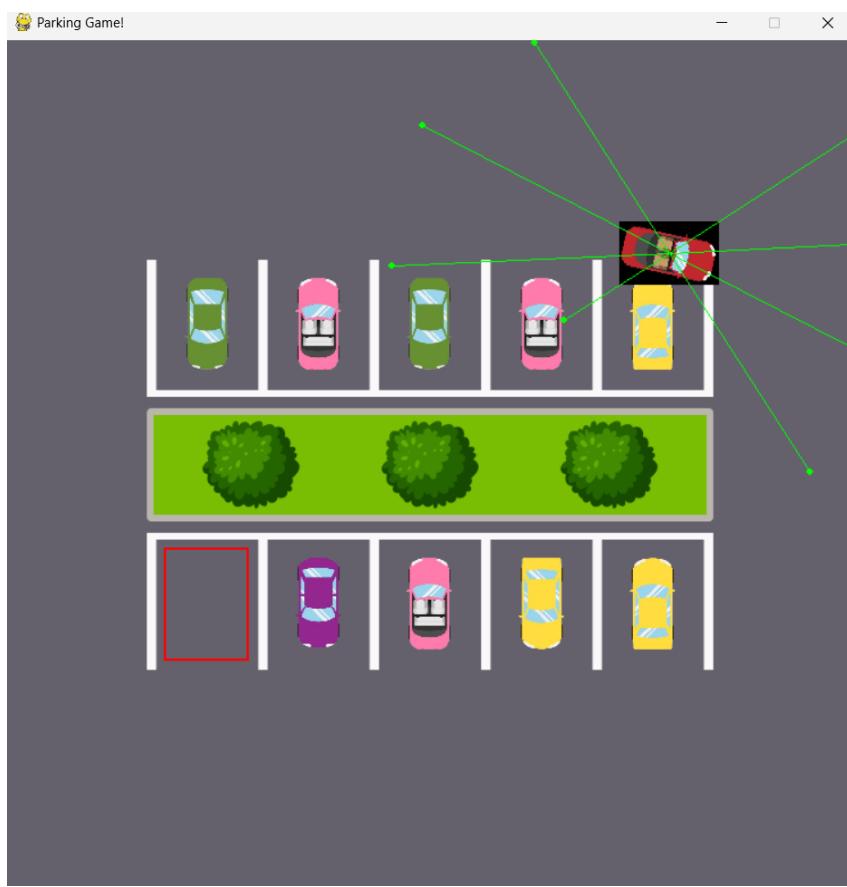


**Εικόνα 4.4.** Παράδειγμα εικόνας για τη δημιουργία μάσκας του χάρτη.

Ωστόσο, το μειονέκτημα αυτής της μεθόδου είναι το υπολογιστικό κόστος που επιβάλλει, παρά τη χρήση μασκών. Εξάλλου, πρόκειται για έναν υπολογισμό ο οποίος θα πρέπει να εκτελείται 20 φορές το δευτερόλεπτο, καθώς το παιχνίδι λειτουργεί στα 20 fps. Στην πράξη, η υπολογιστική επιβάρυνση της τεχνικής αυτής, οδήγησε σε μειωμένη απόδοση του παιχνιδιού. Η λύση σε αυτό το πρόβλημα, δίνεται από την πολύ φθηνότερη μέθοδο της ανίχνευσης σύγκρουσης ορθογωνίων σχημάτων

#### 4.3 Κατασκευή παιχνιδιού

(rectangle collision detection). Συγκεκριμένα, τα αντικείμενα του χάρτη στα οποία μπορεί να προκύψει σύγκρουση (δηλαδή αυτοκίνητα, κήπος, τοίχοι), αναπαρίστανται στο παιχνίδι ως ορθογώνια παραλληλόγραφα, τα οποία καλύπτονται στην οθόνη του παιχνιδιού από τις αντίστοιχες εικόνες. Έτσι, το αυτοκίνητο του πράκτορα αναπαρίσταται από ένα ορθογώνιο παραλληλόγραφο, το οποίο παίζει επίσης, σημαντικό ρόλο στο μηχανισμό της περιστροφής του αυτοκινήτου. Επομένως, μπορεί να πραγματοποιηθεί η ανίχνευση συγκρούσεων με τη μέθοδο των ορθογωνίων σχημάτων, η οποία ελέγχει αν δύο ορθογώνια σχήματα επικαλύπτονται. Ωστόσο, ενδέχεται να προκύψουν λάθη από τη χρήση της μεθόδου. Συγκεκριμένα, το ορθογώνιο του αυτοκινήτου, δεν περιβάλλει πάντα, ακριβώς, την εικόνα του αυτοκινήτου. Πιο αναλυτικά, όταν η εικόνα του αυτοκινήτου βρίσκεται υπό γωνία, τότε το ορθογώνιο του αυτοκινήτου είναι μεγαλύτερο από αυτήν, όπως φαίνεται στην Εικόνα 4.5.

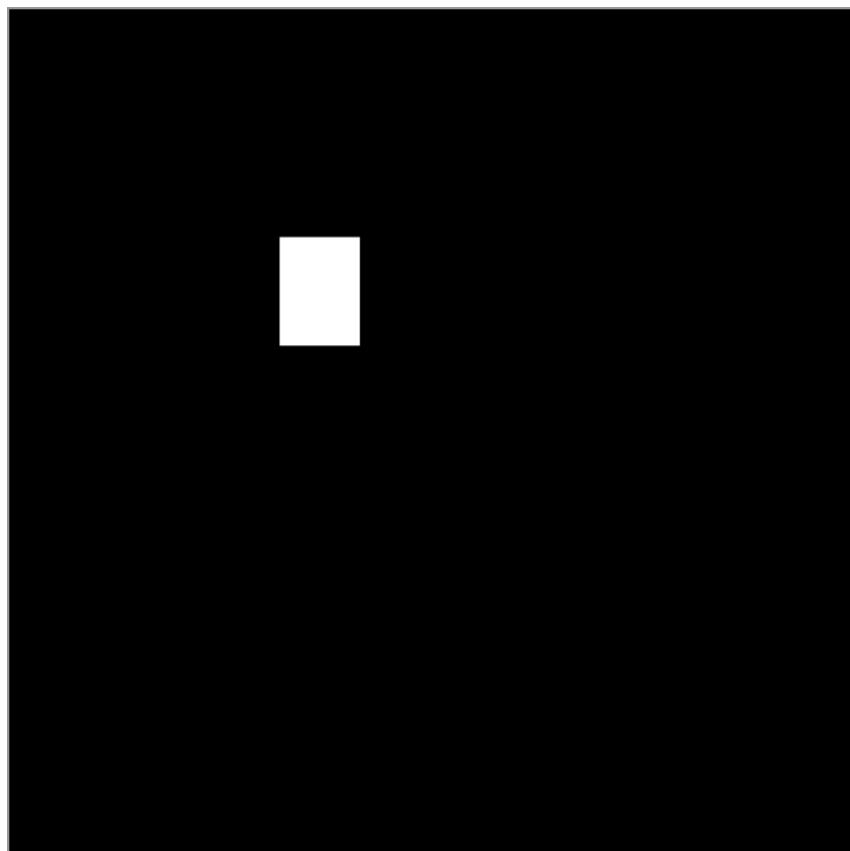


**Εικόνα 4.5.** Ορθογώνιο αυτοκινήτου έναντι εικόνας αυτοκινήτου. Παρατηρούμε πως η ανίχνευση σύγκρουσης με τη μέθοδο των ορθογωνίων σχημάτων μπορεί να είναι ανακριβής.

Το γεγονός αυτό, προκαλεί ψευδώς θετικές συγκρούσεις, δηλαδή περιπτώσεις στις οποίες συγκρούεται μόνο το ορθογώνιο του αυτοκινήτου και όχι η εικόνα του, όπως βλέπουμε στην Εικόνα 4.5. Επομένως, η τελική λύση που επιλέχθηκε, αποτελεί έναν συνδυασμό των δύο μεθόδων. Αρχικά, πραγματοποιείται η ανίχνευση σύγκρουσης με τη μέθοδο των ορθογωνίων σχημάτων, και

εφόσον αυτή επιβεβαιωθεί, τότε εφαρμόζεται η ανίχνευση σύγκρουσης με τη μέθοδο των pixel. Έτσι, εξασφαλίζεται τόσο η ακρίβεια της ανίχνευσης, όσο και η ελαχιστοποίηση του υπολογιστικού κόστους.

Με τις ίδιες τεχνικές πραγματοποιείται και η ανίχνευση της εισόδου του αυτοκινήτου, εντός της ελεύθερης θέσης στάθμευσης. Το ενδιαφέρον που παρουσιάζεται εδώ, έγκειται στο γεγονός πως η μέθοδος των pixel εφαρμόζεται ανάποδα σε σχέση με πριν. Συγκεκριμένα, όπως και πριν, πρώτα ανιχνεύεται η σύγκρουση του ορθογωνίου του αυτοκινήτου με το ορθογώνιο της θέσης στάθμευσης, και εφόσον αυτή επιβεβαιωθεί, τότε εφαρμόζεται η ανίχνευση με τη μέθοδο των pixel. Ωστόσο, τώρα χρειάζονται νέες μάσκες, οι οποίες δημιουργούνται από εικόνες όπως η *Εικόνα 4.6*, για την περίπτωση πάλι, της δεύτερης θέσης στάθμευσης.

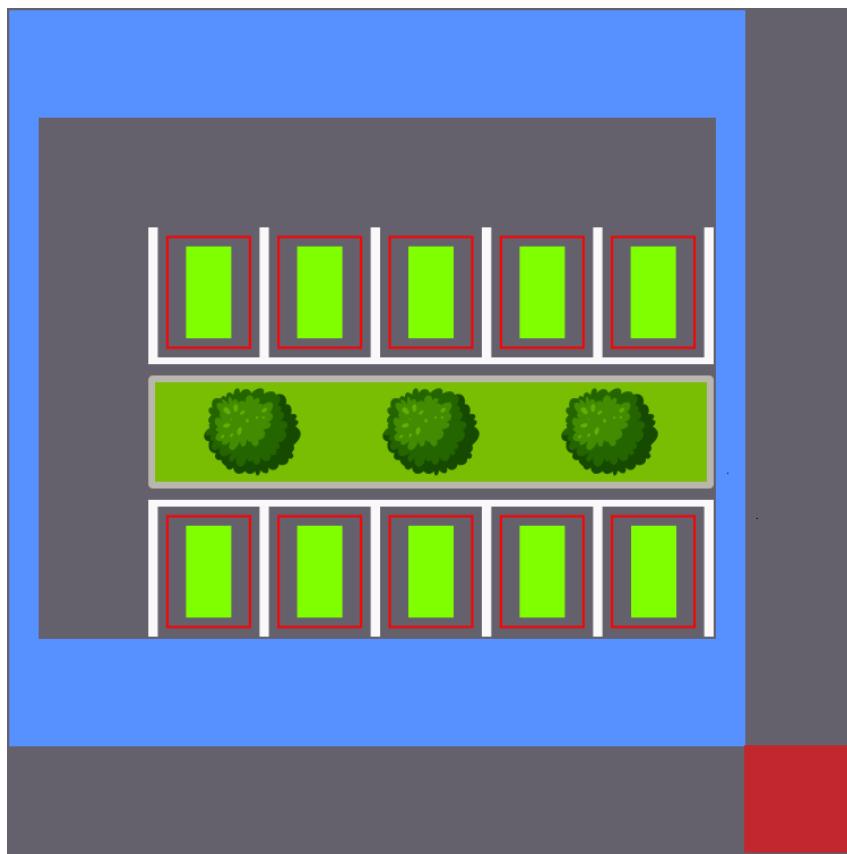


**Εικόνα 4.6.** Παράδειγμα εικόνας για τη δημιουργία μάσκας της ελεύθερης θέσης στάθμευσης.

Όπως γίνεται κατανοητό και από την εικόνα, πλέον ελέγχεται το αν δεν υπάρχει σύγκρουση με την παραπάνω μάσκα, καθώς τότε το αυτοκίνητο έχει εισέλθει ολόκληρο εντός της ελεύθερης θέσης.

### Εκκίνηση από τυχαία θέση

Στη συνέχεια, έχοντας πλέον αναφερθεί στην έννοια του ορθογωνίου του αυτοκινήτου, μπορούμε να σχολιάσουμε την ιδιότητα του πράκτορα να ξεκινάει από τυχαία θέση και γωνία, στην αρχή του κάθε επεισοδίου (*random spawn*). Για να υλοποιηθεί αυτή η ιδιότητα, έπρεπε να καθορίσουμε τις συντεταγμένες (x,y) από τις οποίες θα μπορούσε να ξεκινήσει ο πράκτορας. Ζητούμενο προφανώς είναι, ο πράκτορας να μην ξεκινάει από παράνομη θέση, δηλαδή θέση στην οποία συγκρούεται. Ο υπολογισμός των δυνατών θέσεων εκκίνησης του αυτοκινήτου, έγινε μέσω του Figma και αυτές παρουσιάζονται με γαλάζιο χρώμα στην Εικόνα 4.7.



**Εικόνα 4.7.** Δυνατές θέσεις εκκίνησης του αυτοκινήτου.

Στην παραπάνω εικόνα, τα γαλάζια παραλληλόγραμμα αντιπροσωπεύουν τις δυνατές περιοχές εκκίνησης του αυτοκινήτου, ενώ το κόκκινο παραλληλόγραμμο αντιπροσωπεύει το ορθογώνιο του αυτοκινήτου. Προκειμένου να γίνει κατανοητή η εικόνα, οφείλουν να γίνουν δύο παρατηρήσεις:

- Η πρώτη παρατήρηση σχετίζεται με τις διαστάσεις του ορθογωνίου του αυτοκινήτου. Οι διαστάσεις αυτές, εξαρτώνται από τη γωνία του αυτοκινήτου. Για παράδειγμα, αν η γωνία του αυτοκινήτου είναι  $0^\circ$ , τότε το μήκος του ορθογωνιού θα είναι 82px και το πλάτος του 40px, με

βάση τις διαστάσεις της εικόνας του αυτοκινήτου. Αν όμως η γωνία του αυτοκινήτου είναι  $90^\circ$ , τότε το μήκος και το πλάτος του ορθογωνίου θα είναι ανάποδα. Επειδή η γωνία έναρξης είναι τυχαία, δεν γνωρίζουμε ακριβώς ποιές θα είναι οι διαστάσεις του ορθογωνίου του αυτοκινήτου. Για αυτό, θεωρούμε τη μέγιστη τιμή που μπορούν να πάρουν, δηλαδή 82px. Άρα, το ορθογώνιο του αυτοκινήτου που φαίνεται στην *Εικόνα 4.7*, έχει διαστάσεις  $82 \times 82$ px.

- Η δεύτερη παρατήρηση αφορά την ιδιότητα πως στην Pygame, οι συντεταγμένες ορίζονται με βάση το πάνω αριστερά σημείο ενός σχήματος. Επομένως, τα γαλάζια παραλληλόγραμμα έχουν σχεδιαστεί κατάλληλα, έτσι ώστε όταν το πάνω αριστερά άκρο του ορθογωνίου του αυτοκινήτου βρίσκεται εντός τους, να μην προκύπτει σύγκρουση. Στην *Εικόνα 4.7* παρουσιάζεται μία ακραία περίπτωση αυτού του παραδείγματος, όπου η άκρη του ορθογωνίου του αυτοκινήτου βρίσκεται στην άκρη ενός γαλάζιου παραλληλογράμμου και παρόλα αυτά, δεν προκύπτει κάποια σύγκρουση.

Βέβαια, με αυτόν τον τρόπο περιορίζονται οι δυνατές θέσεις εκκίνησης του αυτοκινήτου, διότι αποκλείονται κάποιοι συνδυασμοί συντεταγμένων και γωνίας εκκίνησης, οι οποίοι δεν θα προκαλούσαν σύγκρουση. Παρόλα αυτά, το πλήθος σημείων αυτών των παραλληλογράμμων, σε συνδυασμό με όλες τις πιθανές γωνίες εκκίνησης του αυτοκινήτου ( $0^\circ$ - $360^\circ$ ), δημιουργούν έναν τεράστιο και ποικιλόμορφο χώρο εκκίνησης, ο οποίος σίγουρα επαρκεί για τη γενίκευση του πράκτορα.

## Αισθητήρες

Προκειμένου ο πράκτορας να αντιλαμβάνεται το περιβάλλον γύρω του, επιλέχθηκε μία από τις δημοφιλέστερες μορφές αισθητήρων για τέτοιου είδους παιχνίδια, οι ακτίνες ραντάρ (*raycasts*). Οι ακτίνες είναι ευθύγραμμες γραμμές, που εκτείνονται από τον πράκτορα προς τα έξω και με βάση το μήκος τους, ο πράκτορας μπορεί να εκτιμήσει τις αποστάσεις με τα αντικείμενα γύρω του. Δεν υπάρχει κάποια σχετική μέθοδος στην Pygame για το μηχανισμό αυτό, επομένως η υλοποίηση των ακτίνων έγινε από την αρχή. Η διαδικασία που ακολουθήθηκε για τη σχεδίαση των ακτίνων, παρουσιάζει ενδιαφέρον ως προς τη μείωση του υπολογιστικού κόστους. Συγκεκριμένα, η σχεδίαση μίας ακτίνας ξεκινάει από το κέντρο του αυτοκινήτου του πράκτορα και συνεχίζει μέχρι να συναντήσει κάποιο εμπόδιο ή έως ότου φτάσει στο μέγιστο μήκος της. Η ανίχνευση των εμποδίων γίνεται ξανά με τη χρήση της μάσκας του χάρτη, ελέγχωντας αν το σημείο στο άκρο της ακτίνας έχει την τιμή 1 στη μάσκα (δηλαδή βρίσκεται αντικείμενο στο σημείο) ή 0 (δηλαδή το σημείο είναι ελεύθερο). Όμως, αποδείχθηκε ότι η αύξηση της ακτίνας κατά 1 pixel τη φορά, προκαλεί πολύ μεγάλο υπολογιστικό κόστος. Αυτό είναι λογικό, αν αναλογιστεί κανείς πως το μέγιστο μήκος της ακτίνας είναι 200px (θα εξηγήσουμε γιατί στη συνέχεια), πως υπάρχουν συνολικά 8 ακτίνες και πως η ανίχνευση αντικειμένων πρέπει να γίνει σε κάθε frame, δηλαδή 20 φορές το δευτερόλεπτο. Το αποτέλεσμα ήταν η εμφανής μείωση της απόδοσης του παιχνιδιού. Επομένως, η λύση που επιλέχθηκε

ήταν η αύξηση της ακτίνας κατά ένα βήμα των 20px τη φορά. Όταν ανιχνευτεί αντικείμενο, τότε το μήκος της ακτίνας μειώνεται κατά 1 επαναληπτικά, έως ότου βρεθεί η αρχή του αντικειμένου. Το μέγεθος του βήματος ορίστηκε στα 20px και όχι πιο μεγάλο, ώστε να αποφευχθεί το πρόβλημα της υπερπήδησης των αντικειμένων από την ακτίνα, το οποίο συνέβαινε με μεγαλύτερα βήματα, όταν η ακτίνα περνούσε από ακμές των αντικειμένων. Επομένως, με τον τρόπο που περιγράψαμε, εξασφαλίζεται η ακρίβεια της ανίχνευσης, ενώ ταυτόχρονα μειώνεται σημαντικά το υπολογιστικό κόστος.

Η λογική πίσω από την επιλογή του μέγιστου μήκους της ακτίνας, ήταν η παροχή έγκαιρης ενημέρωσης στον πράκτορα. Συγκεκριμένα, είναι επιθυμητό, ακόμα και όταν ο πράκτορας κινείται με τη μέγιστη ταχύτητα (6px), η ακτίνα να τον προειδοποιήσει για την ύπαρξη εμποδίου, αρκούντως νωρίς, ώστε αν πατήσει εκεινή τη στιγμή το αντίθετο πλήκτρο (DOWN), να προλάβει να σταματήσει προτού συγκρουστεί. Ο υπολογισμός του απαιτούμενου μήκους για να ισχύει αυτή η συνθήκη, περιγράφεται παρακάτω.

Αρχικά, ας υπολογίσουμε το πλήθος των frames που θα χρειαστεί το αυτοκίνητο για να σταματήσει ( $f$ ), με δεδομένο ότι η αρχική του ταχύτητα είναι  $u_0 = 6\text{px/frame}$ , η τελική του ταχύτητα είναι  $u_f = 0\text{px/frame}$  και η επιτάχυνση του αυτοκινήτου είναι  $a = -0.1\text{px/frame}^2$ . Από την κινηματική, γνωρίζουμε ότι ισχύει η εξίσωση 4.2, για την ταχύτητα του αυτοκινήτου σε κάθε frame  $n$ :

$$u(n) = u_0 + a \cdot n \quad (4.2)$$

Αντικαθιστώντας για  $n = f$  και με βάση τα παραπάνω δεδομένα, προκύπτει ότι  $f = 60 \text{ frames}$ .

Στη συνέχεια, θα υπολογίσουμε την τελική απόσταση  $d$ , που θα διανύσει το αυτοκίνητο, σε αυτά τα 60 frames. Η απόσταση αυτή, δίνεται από το άθροισμα των αποστάσεων που διήνυσε το αυτοκίνητο σε κάθε frame, δηλαδή την ταχύτητα του αυτοκινήτου σε κάθε frame (εξίσωση 4.3):

$$d = \sum_{i=1}^f u(i) = \sum_{i=1}^f (u_0 + a \cdot i) = \sum_{i=1}^{60} (6 - 0.1 \cdot i) \quad (4.3)$$

Αυτό το άθροισμα υπολογίζεται χρησιμοποιώντας τον αντίστοιχο τύπο των αριθμητικών σειρών, που δίνεται στην εξίσωση 4.4:

$$d = \frac{f}{2} \cdot (u_0 + u_f) = \frac{60}{2} \cdot (6 + 0.1) = 183 \text{ pixels} \quad (4.4)$$

Άρα, προκύπτει πως το αυτοκίνητο θα χρειαστεί 183 pixels για να σταματήσει. Ωστόσο, επιλέξαμε να μην υπάρχει ακτίνα κάθετη στο αυτοκίνητο (δηλαδή με γωνία  $90^\circ$ ), για λόγους που θα εξηγηθούν στην υποενότητα 4.3.4. Η κοντινότερη ακτίνα προς την κατεύθυνση της κίνησης του πράκτορα, έχει γωνία  $75^\circ$ , επομένως μπορούμε να υπολογίσουμε το απαραίτητο μήκος αυτής μέσω τριγωνομετρίας,

στο ορθογώνιο τρίγωνο που σχηματίζεται (εξίσωση 4.5):

$$\text{μήκος_ακτίνας} = \frac{\text{μήκος_καθέτου}}{\cos(15^\circ)} = 190 \text{ pixels} \quad (4.5)$$

Επομένως, το απαιτούμενο μήκος των ακτινών μας είναι 190 pixels. Επιλέξαμε την στρογγυλοποίηση της τιμής αυτής στα 200 pixels, ώστε να αφήσουμε ένα μικρό περιθώριο αντίδρασης, στους παίκτες του παιχνιδιού.

## Φιγούρες αυτοκινήτων

Τέλος, μία καθαρά αισθητική παρατήρηση αφορά τις εικόνες των αυτοκινήτων. Συγκεκριμένα, επιλέχθηκαν 4 διαφορετικοί τύποι σταθμευμένων αυτοκινήτων με διαφορετικά χρώματα. Σε κάθε επεισόδιο, η κατανομή των τύπων αυτοκινήτων σε θέσεις είναι τυχαία, αλλά πάντα θα βρίσκεται τουλάχιστον ένα αυτοκίνητο από κάθε τύπο. Με τον τρόπο αυτό, τα επεισόδια διαφέρουν μεταξύ τους και προστίθεται ποικιλία στο παιχνίδι. Ακόμα, δημιουργήθηκαν πολλές παραλλαγές του αυτοκινήτου του πράκτορα, ώστε κάθε αλγόριθμος να έχει διαφορετικό χρώμα. Συνολικά όμως, τα χρώματα όλων των αυτοκινήτων επελέγησαν έτσι, ώστε να διαφέρουν σημαντικά μεταξύ τους και να διακρίνονται εύκολα από όλους τους παίκτες.

### 4.3.4 Αδυναμίες

#### Μηχανισμός σύγκρουσης

Όπως προαναφέρθηκε, είναι γνωστή μία αστοχία του παιχνιδιού, η οποία σχετίζεται με τον μηχανισμό της σύγκρουσης. Συγκεκριμένα, όταν το αυτοκίνητο συγκρουστεί με ένα άλλο αντικείμενο, τότε το παιχνίδι θα το μετακινεί προς την αντίθετη κατεύθυνση επαναλαμβανόμενα, μέχρι να σταματήσει να συγκρούεται. Η αστοχία εμφανίζεται, όταν κατά τη μετακίνηση του αυτοκινήτου προς την αντίθετη κατεύθυνση, προκύψει δεύτερη σύγκρουση, με άλλο αντικείμενο. Τότε, δεν σταματάει η επανάληψη της μετακίνησης του αυτοκινήτου, αλλά συνεχίζεται έως ότου το αυτοκίνητο να μην συγκρούεται με κανένα αντικείμενο. Αυτό μπορεί να συμβεί σε 2 περιπτώσεις, οι οποίες παρουσιάζονται παρακάτω, μαζί με τις μεθόδους αντιμετώπισης που χρησιμοποιήθηκαν:

1. Όταν το αυτοκίνητο βρίσκεται ανάμεσα σε δύο σταθμευμένα οχήματα. Να σημειωθεί πως υπάρχει ικανός χώρος, ώστε ο παίκτης να τοποθετήσει το αυτοκίνητο του μεταξύ δύο άλλων οχημάτων. Τότε, η αστοχία θα παρουσιαστεί, αν το αυτοκίνητο συγκρουστεί με το πρώτο σταθμευμένο όχημα, με τρόπο τέτοιο, ώστε η αναπήδηση του προς τα πίσω να το ωθήσει να συγκρουστεί με το δεύτερο σταθμευμένο όχημα. Σε αυτήν την περίπτωση, το αυτοκίνητο θα εμφανιστεί απότομα στον παίκτη πίσω από το δεύτερο σταθμευμένο όχημα. Αυτή η εξέλιξη,

παρόλο που δεν είχε σχεδιαστεί, εντούτοις δεν είναι πλήρως ανεπιθύμητη. Συγκεκριμένα, μπορεί να θεωρηθεί ως χαρακτηριστικό του παιχνιδιού, προκειμένου να απεγκλωβίζει το αυτοκίνητο του παίκτη, όταν αυτό έχει κολλήσει στον στενό χώρο μεταξύ δύο άλλων οχημάτων.

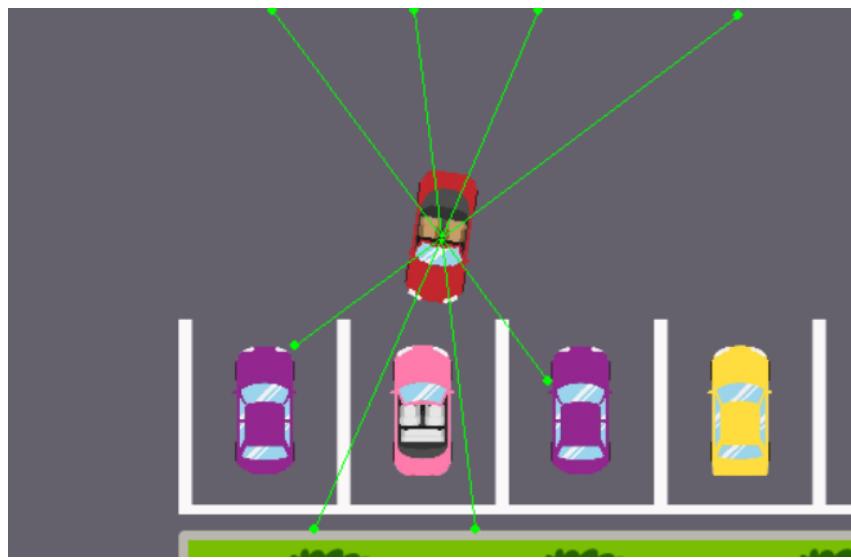
2. Όταν το αυτοκίνητο βρίσκεται υπό γωνία στις 4 άκρες του χάρτη. Συγκεκριμένα, περιμετρικά του χάρτη υπάρχει ένας λεπτός τοίχος, ώστε να εμποδίζει το αυτοκίνητο από το να βγαίνει εκτός των ορίων του. Έτσι, μπορεί να δημιουργηθεί η ίδια συνθήκη με πριν, όπου για παράδειγμα, το ρόλο του πρώτου οχήματος παίζει ένας κατακόρυφος τοίχος και το ρόλο του δεύτερου οχήματος παίζει ένας οριζόντιος τοίχος. Ωστόσο, αυτή η περίπτωση είναι πιο σοβαρή, καθώς μετά τη μετακίνηση του, το αυτοκίνητο καταλήγει εκτός χάρτη. Για αυτό τότε, επιλέξαμε το αυτοκίνητο να μεταφέρεται αμέσως εντός του χάρτη, σε τυχαία θέση και γωνία.

Προφανώς, οι εξελίξεις των παραπάνω σεναρίων δεν αντικατοπτρίζουν τον πραγματικό κόσμο. Ωστόσο, δεν είναι καταστροφικά για το παιχνίδι, αφού ακόμα και στην σπάνια περίπτωση όπου συμβούν οι περιπτώσεις αυτές, το επεισόδιο μπορεί ύστερα να συνεχιστεί κανονικά. Γενικότερα, ο μηχανισμός της σύγκρουσης παρουσιάσει αρκετά προβλήματα, εκ των οποίων η επίλυση του ενός, συχνά προκαλούσε τη δημιουργία του άλλου. Επομένως αυτή η υλοποίηση, θεωρήθηκε η καλύτερη δυνάτη, με δεδομένο ότι βασίζεται πλήρως στη βιβλιοθήκη Pygame. Για καλύτερα αποτελέσματα, θα μπορούσε κάποιος να στραφεί σε άλλες βιβλιοθήκες, εξειδικευμένες στην προσομοίωση μηχανισμών φυσικής όπως π.χ. η Pymunk.

### Μηχανισμός αισθητήρων

Ο μηχανισμός των αισθητήρων που υλοποιήθηκε, δηλαδή η τεχνική raycasting, είναι αρκετά απλοϊκή και μπορεί να αποδειχθεί ανεπαρκής, σε ορισμένες περιπτώσεις. Συγκεκριμένα, μπορεί ένα αντικείμενο να είναι αρκετά λεπτό ή υπό κατάλληλη γωνία, ώστε να μην ανιχνευτεί από τις ακτίνες του πράκτορα. Προσπαθήσαμε να περιορίσουμε αυτό το φαινόμενο, επιλέγοντας κατάλληλες γωνίες ακτίνων, ώστε αυτές να παρέχουν πληροφορία για αντικείμενα που βρίσκονται προς τα μπροστά ή προς τα πίσω, σε σχέση με το αυτοκίνητο. Επομένως, πυκνώσαμε τις ακτίνες στις κατευθύνσεις στις οποίες, το αυτοκίνητο μπορεί να κινηθεί. Ωστόσο, προβλήματα αυτού του είδους μπορούν ακόμα να εμφανιστούν, όπως αποδεικνύει η Εικόνα 4.8.

Μία προφανής λύση σε αυτό το ζήτημα είναι η αύξηση του πλήθους των ακτίνων. Όμως αυτό δεν επιλέχθηκε, πρώτον για λόγους διατήρησης του υπολογιστικού φορτίου του παιχνιδιού σε χαμηλά επίπεδα και δεύτερον για λόγους μείωσης της πολυπλοκότητας του νευρωνικού δικτύου του πράκτορα. Συγκεκριμένα, η αύξηση του πλήθους των ακτίνων θα σήμαινε αύξηση των εισόδων του δικτύου, το οποίο θα οδηγούσε σε πιο αργή και δύσκολη εκπαίδευση (Dey 2023). Άλλωστε, οι περισσότερες εργασίες που κάνουν χρήση της τεχνικής raycasting, χρησιμοποιούν παρόμοιο αριθμό ακτίνων και επιβεβαιώθηκε στην πράξη, μέσω της εκπαίδευσης, πως οι πράκτορες δεν



**Εικόνα 4.8.** Παράδειγμα ανεπάρκειας των αισθητήρων. Παρατηρούμε ότι δεν ανισχνεύεται το ροζ αυτοκίνητο, παρόλο που βρίσκεται εντός του βεληνεκούς των ακτίνων.

δυσκολεύτηκαν να μάθουν να αποφεύγουν τις συγκρούσεις. Επομένως, παραδείγματα αστοχιών ανίχνευσης όπως αυτό της εικόνας 4.8, θεωρούνται αμελητέα, για την επίτευξη του στόχου της εργασίας,



## 5 Εκπαίδευση

Η εκπαίδευση αποτελεί τη διαδικασία, στην οποία οι πράκτορες ενισχυτικής μάθησης αλληλεπιδρούν με το περιβάλλον και μαθαίνουν να λύνουν το πρόβλημα, για το οποίο έχουν σχεδιαστεί.

Η εκπαίδευση πρακτόρων αποτελεί τον πυρήνα αυτής της εργασίας. Διεξήχθησαν εκπαιδεύσεις με όλους τους αλγορίθμους που που μελετήσαμε θεωρητικά στο Κεφάλαιο 3 και δοκιμάστηκαν διάφορες μέθοδοι και τεχνικές, ώστε να πετύχει ο κάθε αλγόριθμος την καλύτερη δυνατή απόδοση στο παιχνίδι. Η διαδικασία αυτή χρειάστηκε αρκετούς μήνες και κάτα τη διάρκεια της, προέκυψαν πολλές δυσκολίες και προβλήματα. Ταυτόχρονα όμως, συγκεντρώθηκαν ουσιαστικές πληροφορίες και προέκυψαν χρήσιμα συμπεράσματα.

Το παρόν κεφάλαιο, προσπαθεί να οργανώσει με τον βέλτιστο τρόπο, την αχανή αυτή διαδικασία και να αναδείξει τις σημαντικότερες πτυχές της. Έτσι, το κεφαλαίο της εκπαίδευσης διαρθρώνεται ως εξής:

- Στην **Ενότητα 5.1**, παρατίθενται αρχικά, οι κύριοι στόχοι της εκπαίδευσης πρακτόρων.
- Στην **Ενότητα 5.2**, αναλύονται τα εργαλεία που χρησιμοποιήθηκαν και τα πλεονεκτήματα που προσέφερε το καθένα.
- Στην **Ενότητα 5.3**, γίνεται αναφορά στους υπολογιστικούς πόρους που χρησιμοποιήθηκαν για την εκτέλεση των εκπαιδεύσεων.
- Στην **Ενότητα 5.4**, παρουσιάζονται κάποια συνολικά στατιστικά της διαδικασίας εκπαίδευσης.
- Στην **Ενότητα 5.5**, εξετάζονται τα προβλήματα της ενισχυτικής μάθησης και σχολιάζεται η εμφάνισή τους, στις εκπαιδεύσεις που διεξήχθησαν.
- Στην **Ενότητα 5.6**, δίνονται κάποιες γενικές, καλές πρακτικές για την εκπαίδευση πρακτόρων, καθώς και πιο συγκεκριμένες συμβουλές, βασισμένες στην εμπειρία που αποκτήθηκε κατά τη διάρκεια των εκπαιδεύσεων.
- Στην **Ενότητα 5.7**, αναλύεται η τελική μοντελοποίηση του συστήματος, όσον αφορά την επιλογή της αρχιτεκτονικής των νευρωνικών δικτύων και τη σχεδίαση της συνάρτησης ανταμοιβής, η οποία οδήγησε στα καλύτερα αποτελέσματα των πρακτόρων.
- Στις **Ενότητες 5.8, 5.9, 5.10, 5.11** και **5.12**, εξετάζονται οι εκπαιδεύσεις του κάθε αλγορίθμου ξεχωριστά, κάνοντας πρώτα, μία επισκόπηση του συνόλου των εκπαιδεύσεων του αλγορίθμου και έπειτα μελετώντας την καλύτερη εκπαίδευση του.

## 5.1 Στόχοι

Οι βασικοί στόχοι, τους οποίους προσπαθούν να επιτυχούν οι αλγόριθμοι ενισχυτικής μάθησης, μέσω της διαδικασίας της εκπαίδευσης, χωρίζονται σε δύο κατηγορίες: στην αποδοτικότητα τους και στην τελική επίδοση τους.

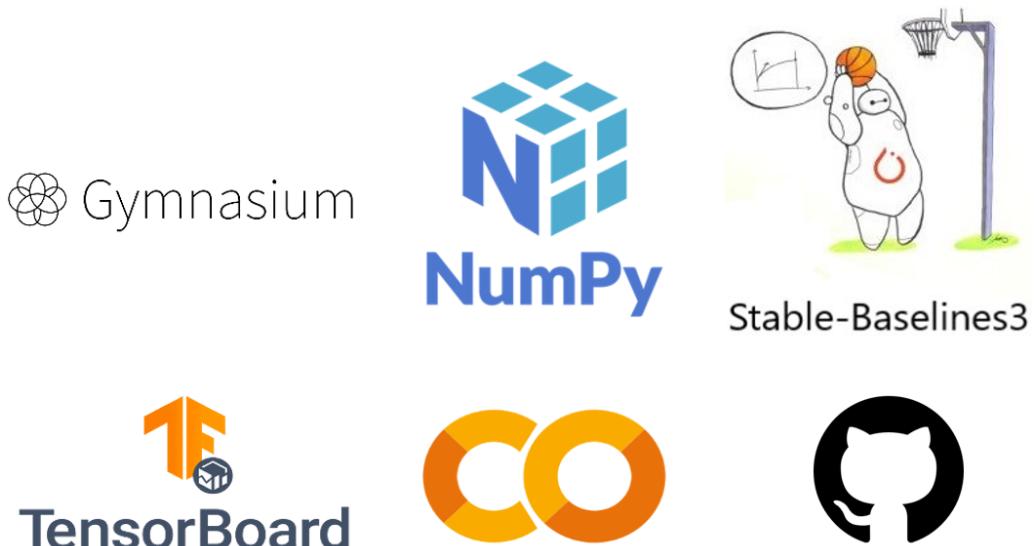
Αρχικά, η αποδοτικότητα των αλγορίθμων ενισχυτικής μάθησης, αναφέρεται στον χρόνο εκπαίδευσης. Στόχος των αλγορίθμων αυτών, είναι να μαθαίνουν τη βέλτιστη πολιτική, σε περιορισμένο αριθμό αλληλεπιδράσεων με το περιβάλλον. Τότε, οι αλγόριθμοι χαρακτηρίζονται **αποδοτικοί** ως προς τη χρήση δειγμάτων (*sample efficient*) και επιτυγχάνουν τη μείωση του χρόνου εκπαίδευσης.

Η τελική επίδοση των αλγορίθμων, αφορά τον βαθμό στον οποίο καταφέρνουν να εκτελέσουν την εργασία, για την οποία εκπαιδεύτηκαν. Σημαντικές παράμετροι αυτού του στόχου, αποτελούν η **ευστάθεια** (*stability*) και **σύγκλιση** (*convergence*) της εκπαίδευσης. Συγκεκριμένα, η ευστάθεια περιγράφει την ομαλή αύξηση της καμπύλης μάθησης, χωρίς την ύπαρξη μεγάλων αυξομειώσεων. Η σύγκλιση αναφέρεται στην ευθυγράμμιση της καμπύλης μάθησης προς το τέλος της εκπαίδευσης, όπου ο αλγορίθμος σταθεροποιεί πλήρως την πολιτική του. Προφανώς, είναι επιθυμητό η πολιτική στην οποία συγκλίνει ο αλγόριθμος να είναι η βέλτιστη, ή κοντά σε αυτήν. Τέλος, μέσα από την εκπαίδευση, πρέπει να διασφαλιστεί η **γενίκευση** (*generalization*) του αλγορίθμου, δηλαδή η ικανότητα του να αποδίδει εξίσου καλά σε σενάρια που δεν έχει δει αυτούσια, κατά τη διάρκεια της εκπαίδευσης. Έτσι, όταν σε διαφορετικές αξιολογήσεις του αλγορίθμου, η επίδοση του είναι συνεπής, τότε τα αποτελέσματα της μάθησης είναι αξιόπιστα και χαρακτηρίζονται από **προβλεψιμότητα** (*predictability*). Μάλιστα, όταν οι διαφορετικές αυτές αξιολογήσεις, γίνονται υπό μικρές αλλαγές στο περιβάλλον ή στις αρχικές συνθήκες του πράκτορα, τότε ο αλγόριθμος διακατέχεται από **ανθεκτικότητα** (*robustness*).

Είναι σημαντικό να ελέγχεται τακτικά κατά τη διάρκεια της εκπαίδευσης, κατά πόσο ικανοποιούνται οι παραπάνω στόχοι, ώστε να γίνουν οι κατάλληλες προσαρμογές, εφόσον αυτές απαιτούνται.

## 5.2 Εργαλεία

Τα εργαλεία που χρησιμοποιήθηκαν για τις εκπαιδεύσεις των πρακτόρων παρουσιάζονται στην *Εικόνα 5.1*. Στη συνέχεια, ακολουθεί μία σύντομη παρουσίαση του καθενός, καθώς και περιγραφή του τρόπου με τον οποίο αξιοποιήθηκε στην εργασία.



**Εικόνα 5.1.** Εργαλεία εκπαίδευσης.

### 5.2.1 Gymnasium

Η βιβλιοθήκη [OpenAI Gymnasium](#), έχει αναλυθεί στην *Ενότητα 4.1* ως ένα σημαντικό εργαλείο για την ανάπτυξη και σύγκριση αλγορίθμων ενισχυτικής μάθησης. Πέραν όμως, από τη μεγάλη ποικιλία περιβαλλόντων και μεθόδων που πάρεχει η βιβλιοθήκη, το μεγαλύτερο πλεονέκτημα που προσφέρει η χρήση της, είναι η δυνατότητα εκτέλεσης του περιβάλλοντος χωρίς γραφικά (*renderless*). Αυτό είναι ιδιαίτερα χρήσιμο στη διαδικασία της εκπαίδευσης, καθώς μειώνει σε μεγάλο βαθμό το υπολογιστικό κόστος και επιταχύνει τη διάρκεια του κάθε επεισοδίου. Για παράδειγμα στην περίπτωση μας, ο πραγματικός χρόνος ενός επεισοδίου του παιχνιδιού είναι 30s, ενώ ο χρόνος εκτέλεσης του περιβάλλοντος χωρίς γραφικά μετρήθηκε στα 0.13s. Επομένως, χρησιμοποιώντας αυτή την ιδιότητα της βιβλιοθήκης OpenAI Gymnasium, η εκπαίδευση γίνεται 231 φορές πιο γρήγορη.

Ωστόσο, μπορεί να απαιτείται προσαρμογή διαφόρων παραμέτρων του παιχνιδιού για εκτέλεση χωρίς γραφικά. Για παράδειγμα, πρέπει να αποφευχθούν χρονικές συνθήκες εντός του παιχνιδιού, επειδή αυτό πλέον, θα εκτελείται πολύ πιο γρήγορα. Έτσι, ένα λάθος που κάναμε στις πρώτες εκπαιδεύσεις, ήταν η χρήση της χρονικής συνθήκης των 2 δευτερολέπτων για την επιτυχή στάθμευση του πράκτορα. Προφανώς, αυτό δεν ήταν πλέον εφικτό, καθώς το κάθε επεισόδιο είχε πλέον διάρκεια περίπου 0.13s. Για αυτό, προσαρμόσαμε τη συνθήκη αυτή, ώστε να ελέγχεται το πλήθος των βημάτων, στα οποία ο πράκτορας βρίσκεται ακίνητος εντός της θέσης στάθμευσης και όχι ο αντίστοιχος χρόνος.

### 5.2.2 Numpy

Η [Numpy](#) είναι μία δημοφιλής βιβλιοθήκη της Python για επιστημονικούς υπολογισμούς. Βασικό της χαρακτηριστικό αποτελεί η υποστήριξη για μεγάλους, πολυδιάστατους πίνακες (*arrays*), καθώς και η παροχή διαφορών μαθηματικών συναρτήσεων για την αποδοτική επεξεργασία αυτών των πινάκων. Επομένως, η Numpy χρησιμοποιήθηκε σε αυτήν την εργασία, για την αποθήκευση και την ενημέρωση του πίνακα  $Q$ .

### 5.2.3 Stable-Baselines3

Η [Stable-Baselines3](#) είναι μία βιβλιοθήκη της Python, που χρησιμοποιείται για την εκπαίδευση πρακτόρων ενισχυτικής μάθησης. Συγκεκριμένα, αποτελεί μία συλλογή αξιοπίστων υλοποιήσεων, σύγχρονων αλγορίθμων ενισχυτικής μάθησης. Αξίζει να τονιστεί η αξιοπιστία των υλοποιήσεων αυτών, αφού τα αποτελέσματα τους έχουν συγκριθεί με αυτά από τις αρχικές δημοσιεύσεις των αλγορίθμων. Αυτό είναι πολύ σημαντικό, καθώς, όπως αναφέρεται και στο (Amid 2018), η ενισχυτική μάθηση είναι αρκετά ασταθής, με πολλούς παράγοντες που μπορεί να προκαλέσουν την αποτυχία μίας εκπαίδευσης. Επομένως, χρησιμοποιώντας μία υλοποίηση αλγορίθμου που έχει αποδειχθεί ότι λειτουργεί, αφαιρείται μία πιθανή πηγή σφάλματος.

Ακόμα, η βιβλιοθήκη είναι γνωστή για τη φιλική προς την χρήστη διεπαφή της. Θα μπορούσαμε να πούμε πως αποτελεί την αντίστοιχη βιβλιοθήκη της SciKit Learn, για τον τομέα της ενισχυτικής μάθησης, αφού και οι δύο βιβλιοθήκες διακρίνονται για την ευκολία χρήσης τους. Επομένως, η Stable-Baselines3, επιλέχθηκε σε αυτήν την εργασία έναντι άλλων, δημοφιλών επιλογών όπως η PyTorch ή το TensorFlow, διότι θεωρείται η ιδανική επιλογή, για αρχάριους στον τομέα της ενισχυτικής νοημοσύνης και για εργασίες, όπου ο κύριος στόχος είναι η εφαρμογή υπαρχόντων αλγορίθμων, παρά η ανάπτυξη νέων.

Κατά την εκπαίδευση των πρακτόρων, η βιβλιοθήκη παρέχει τη δυνατότητα αποθήκευσης του πράκτορα σε ένα αρχείο `.zip`, ανά ένα καθορισμένο αριθμό βημάτων. Αυτό το αρχείο `.zip` δεν περιέχει απλώς τα βάρη του νευρωνικού δικτύου, αλλά και την αρχιτεκτονική του, τις ρυθμίσεις του αλγορίθμου, καθώς και πληροφορίες του περιβάλλοντος. Με αυτόν τον τρόπο, εξασφαλίζεται ότι ο εκπαιδευμένος πράκτορας μπορεί να φορτωθεί ξανά στο μέλλον για συνέχιση της εκπαίδευσης του ή για αξιολόγηση. Το μέγεθος του αρχείου εξαρτάται από το μέγεθος του νευρωνικού δικτύου, καθώς και τον αλγόριθμο εκπαίδευσης. Στην περίπτωση μας, υπό σταθερή αρχιτεκτονική δικτύου, προέκυψαν τα εξής μεγέθη αρχείων για τους αλγορίθμους που χρησιμοποιήθηκαν, τα οποία αποτελούν μία καλή ένδειξη της πολυπλοκότητας και του κόστους κάθε αλγορίθμου:

- PPO: 0.156 MB
- SAC: 3.054 MB
- DDPG: 3.985 MB

- TD3: 5.971MB

Επιπλέον, αξίζει να σημειωθεί πως η βιβλιοθήκη Stable-Baselines3 χρησιμοποιεί διανυσματοποιημένα περιβάλλοντα (*vectorized environments*). Τα διανυσματοποιημένα περιβάλλοντα, είναι μία μέθοδος που επιτρέπει την εκπαίδευση ενός πράκτορα ενισχυτικής μάθησης σε πολλαπλά ανεξάρτητα περιβάλλοντα, ταυτόχρονα. Έτσι, αξιοποιούνται καλύτερα πόροι του υλικού όπως οι πολυπύρηνοι επεξεργαστές και επιταχύνεται η διαδικασία της εκπαίδευσης. Υπάρχουν δύο ειδών διανυσματοποιημένα περιβάλλοντα στη βιβλιοθήκη Stable-Baselines3:

- To DummyVecEnv, το οποίο εκτελεί τα περιβάλλοντα σειριακά και
- To SubprocVecEnv, το οποίο εκτελεί τα περιβάλλοντα παράλληλα, σε διαφορετικές διεργασίες.

Στην περίπτωση μας, έγινε προσπάθεια για αξιοποίηση του περιβάλλοντος SubprocVecEnv, προκειμένου να αυξηθεί η απόδοση της εκπαίδευσης. Ωστόσο, αυτό δεν κατέστη δυνατό, λόγω της χρήσης της βιβλιοθήκης Pygamer για την ανάπτυξη του παιχνιδιού. Συγκεκριμένα, η Pygamer δεν είναι συμβατή με τη διαδικασία της σειριοποίησης των αντικειμένων της, που απαιτείται για την αποστολή τους σε διαφορετικές διεργασίες. Επομένως, χρησιμοποιήθηκε το απλούστερο περιβάλλον DummyVecEnv, το οποίο παρά το όνομα του, είναι προτιμότερο σε απλά περιβάλλοντα, όπου ο φόρτος της πολυδιεργασίας υπερισχύει του χρόνου εκτέλεσης του περιβάλλοντος.

Τέλος, είναι εξαιρετικά χρήσιμη η εύκολη ενσωμάτωση της βιβλιοθήκης Stable-Baselines3 με τη βιβλιοθήκη OpenAI Gymnasium, καθώς και με το εργάλειο TensorBoard, το οποίο είναι το επόμενο εργαλείο που θα αναλυθεί.

#### 5.2.4 TensorBoard

Το [Tensorboard](#) είναι ένα εργαλείο οπτικοποίησης που χρησιμοποιείται μαζί με διάφορες βιβλιοθήκες μηχανικής μάθησης, για την παρακολούθηση της εκπαίδευσης. Προσφέρει τη σχεδίαση γραφημάτων διαφόρων χρήσιμων μετρικών, όπως της μέσης ανταμοιβής του πράκτορα ανά επεισόδιο (*ep\_rew\_mean*), του μέσου μήκους επεισοδίου (*ep\_len\_mean*), του σφάλματος της πολιτικής (*policy\_loss*), του ρυθμού μάθησης (*learning\_rate*) κ.ά. Αυτό είναι ιδιαίτερα χρήσιμο κατά τη διάρκεια της εκπαίδευσης, καθώς επιτρέπει την παρακολούθηση της εξέλιξης του πράκτορα και την ανίχνευση προβλημάτων. Μάλιστα, τα γραφήματα ανανεώνονται αυτόματα ανά τακτά χρονικά διαστήματα, ενώ προσφέρεται η δυνατότητα σύγκρισης των μετρικών μεταξύ διαφορετικών εκπαιδεύσεων. Έτσι, βοηθάει τους σχεδιαστές συστημάτων μηχανικής μάθησης, να αναγνωρίσουν πιο γρήγορα, τυχόν σφάλματα της εκπαίδευσης και να κατανοήσουν καλύτερα τις αιτίες τους.

Στην περίπτωση μας, οι μετρικές της μέσης ανταμοιβής και του μέσου ποσοστού επιτυχίας του πράκτορα ήταν οι πιο χρήσιμες, για την παρακολούθηση της πορείας της εκπαίδευσης και βοήθησαν σε μεγάλο βαθμό στη βελτίωση της απόδοσης των πρακτόρων. Ακόμα, αξιοποιήθηκε το

## 5.2 Εργαλεία

---

χαρακτηριστικό του Tensorboard που επιτρέπει τη ρύθμιση του βαθμού εξομάλυνσης των γραφικών (*smoothing*). Με τον τρόπο αυτό, οι γραφικές παραστάσεις των μετρικών γίνονται πιο ευανάγνωστες και παρέχεται καλύτερη πληροφορία για την κλίση τους, χωρίς να αποπροσανατολίζεται κάποιος από τις συχνές βυθίσεις και μέγιστα.

### 5.2.5 Colaboratory

Το [Colaboratory](#) (ή *Colab*) είναι μία δωρεάν πλατφόρμα νέφους (*cloud*) της Google, για την εκτέλεση κώδικα Python σε περιβάλλον τύπου Jupyter notebook. Παρέχει πρόσβαση σε ισχυρές υπολογιστικές μονάδες, όπως οι μονάδες επεξεργασίας γραφικών (GPU) και οι μονάδες επεξεργασίας τανυστών (TPU), καθιστώντας το έτσι ιδανικό, για εργασίες μηχανικής και βαθιάς μάθησης. Το όνομα της πλατφόρμας παραπέμπει στη συνεργασία μεταξύ των ερευνητών, η οποία γίνεται δυνατή μέσω των χαρακτηριστικών της συνεργατικής επεξεργασίας σημειωμάτων Jupyter, καθώς και της αποθήκευσης τους στο Google Drive.

Ωστόσο, στη δωρεάν έκδοση του Colab, η πρόσβαση σε ακριβούς πόρους όπως οι GPU είναι είτε ανύπαρκτη ή πολύ περιορισμένη, ενώ τίθενται χρονικά όρια στη διάρκεια εκτέλεσης (*runtime*) των σημειωμάτων. Συγκεκριμένα, ο κάθε χρήστης έχει στη διάθεση του καθημερινά, κάποιες υπολογιστικές μονάδες (*compute units*), οι οποίες αποτελούν μία μορφή συναλλάγματος. Όσο πιο ακριβό το υλικό που χρησιμοποιεί ο χρήστης, τόσο γρηγορότερα εξαντλούνται οι μονάδες του. Μέσω μίας μηνιαίας συνδρομής, ο χρήστης μπορεί να αγοράσει περισσότερες υπολογιστικές μονάδες, να αποκτήσει πρόσβαση σε καλύτερους πόρους και να έχει μεγαλύτερη διάρκεια εκτέλεσης των σημειωμάτων του.

Στην περίπτωση μας, κρίθηκε αναγκαία η αγορά μίας μηνιαίας συνδρομής Colab Pro, η οποία κοστίζει 11.45€, καθώς χρησιμοποιώντας τη δωρεάν έκδοση, η χρήση της GPU κατανάλωνε όλες τις ημερήσιες υπολογιστικές μονάδες σε μόλις μία ώρα. Το πρόβλημα όμως είναι, πως η ενισχυτική μάθηση είναι ιδιαίτερα μη αποδοτική στη χρήση πόρων, με τις εκπαιδεύσεις να χρειάζονται πολύ χρόνο για να ολοκληρωθούν. Έτσι, οι εξαγορασμένες υπολογιστικές μονάδες εξαντλήθηκαν σε μόλις μία εβδομάδα. Επομένως, παρόλο που η πλατφόρμα μας επέτρεψε τη διεξαγωγή εκπαιδεύσεων σε καλύτερα μηχανήματα, η πρόσβαση σε αυτά δεν διήρκησε πολύ. Το πρόβλημα των ακριβών υπηρεσίων νέφους εντοπίζεται και στο άρθρο του (Amid 2018), όπου ο συγγραφέας περιγράφει τις εμπειρίες και τα συμπεράσματα του, από την εκπόνηση ενός έργου ενισχυτικής μάθησης, αναφέροντας χαρακτηριστικά πως ξόδεψε περίπου 850\$ σε πλατφόρμες νέφους, σε διάστημα 8 μηνών, για τις ανάγκες της εκπαίδευσης.

### 5.2.6 Github

Το **Github** είναι μία πλατφόρμα ελέχου έκδοσης λογισμικού ανάπτυξης. Χρησιμοποιείται από προγραμματιστές για την αποθήκευση και διαχείριση του κώδικα τους σε αποθετήρια (*repositories*), την παρακολούθηση των αλλαγών που γίνονται σε αυτόν και τη συνεργασία με άλλους προγραμματιστές σε έργα λογισμικών. Το Github αξιοποιήθηκε στην περίπτωση μας για τη δημιουργία ενός αποθετηρίου αποθήκευσης του κώδικα, καθώς και των υπολοίπων αρχείων της εργασίας. Αποδείχθηκε εξαιρετικά χρήσιμο, καθώς, όπως θα δούμε και στην υποενότητα 5.3 του υλικού, χρησιμοποιήθηκαν διαφορετικοί υπολογιστές για την εκπαίδευση των πρακτόρων. Έτσι, το Github απλοποίησε την επικοινωνία μεταξύ τους, επιτρέποντας την αποθήκευση όλων των αρχείων σε ένα κοινό σημείο αναφοράς και διευκολύνοντας τον συγχρονισμό τους.

## 5.3 Υπολογιστικοί πόροι

Προκειμένου να μειωθεί η συνολική διάρκεια της διαδικασίας εκπαίδευσεων, κρίθηκε απαραίτητη η παραλληλοποίηση της σε διαφορετικούς υπολογιστικούς πόρους. Έτσι, εκτελούσαμε ταυτόχρονα, εκπαίδευσεις διαφορετικών αλγορίθμων, στα υπολογιστικά συστήματα που παρουσιάζονται στον Πίνακα 5.1:

**Πίνακας 5.1:** Πίνακας τεχνικών χαρακτηριστικών διαθέσιμων υπολογιστικών συστημάτων

Σύστημα	Επεξεργαστής (CPU)	Κάρτα γραφικών (GPU)	Μνήμη (RAM)
Laptop HP 15-da1015nv	Intel®Core™i5-8265U	NVIDIA GeForce MX110	8GB
Laptop HP 15-bw009nv	AMD Quad-Core A12-9720P	AMD Radeon™ 530	6GB
Colab Virtual Machine	Intel®Xeon®	Tesla T4	8GB

Είναι γνωστό, πως η εκπαίδευση αλγόριθμών μηχανικής μάθησης επιταχύνεται, όταν χρησιμοποιούνται η GPU, καθώς προσφέρει σημαντικές βελτιώσεις στην υπολογιστική απόδοση, με δυνατότητες παράλληλης επεξεργασίας. Πιο συγκεκριμένα, οι αλγόριθμοι βαθίας ενισχυτικής μάθησης βασίζονται σε νευρωνικά δίκτυα, τα οποία εκπαιδεύονται με μεγάλο αριθμό πολλαπλασιασμών πινάκων και άλλων λειτουργιών που επωφελούνται από την επιτάχυνση της GPU.

Για αυτό, κατά την εκπαίδευση στην πλατφόρμα Colab, επιλέξαμε να χρησιμοποιήσουμε μία από τις διαθέσιμες GPU, την Tesla T4. Αντίστοιχα, κατά την εκπαίδευση στα τοπικά μηχανήματα μας, προσπαθήσαμε να χρησιμοποιήσουμε τις κάρτες γραφικών τους. Αυτό κατέστη εφικτό στην περίπτωση του Laptop HP 15-da1015nv, αφού η GPU του είναι συμβατή με την πλατφόρμα CUDA της

## 5.4 Στατιστικά

---

NVidia<sup>1</sup>, το οποίο αποτέλει προϋπόθεση για χρήση της GPU από τη βιβλιοθήκη Stable-Baselines3. Επομένως, μετά την εγκατάσταση του CUDA Toolkit, της βιβλιοθήκης cuDNN της NVidia και της έκδοσης της PyTorch (βιβλιοθήκης στην οποία είναι γραμμένη η Stable-Baselines3) με υποστήριξη GPU, ήταν δυνατή η εκπαίδευση των αλγορίθμων στη GPU του Laptop HP 15-da1015nv. Αντίθετα, στην περίπτωση του Laptop HP 15-bw009nv, η GPU του δεν υποστηρίζει το CUDA Toolkit, με αποτέλεσμα να μην είναι δυνατή η χρήση της και ως εκ τούτου, χρησιμοποιήθηκε η CPU του για την εκπαίδευση των αλγορίθμων.

Στην πράξη, επιβεβαιώσαμε την επιτάχυνση της εκπαίδευσης χρησιμοποιώντας την GPU, στους υπολογιστικά ακριβότερους αλγορίθμους, της κατηγορίας δράστη-κριτή. Για παράδειγμα, για το Laptop HP 15-da1015nv και υπό τις ίδιες παραμέτρους εκπαίδευσης, τα 1M βήματα εκπαίδευσης του αλγορίθμου TD3 χρειάστηκαν 10.7h με χρήση CPU και 6.03h με χρήση GPU, δηλαδή η εκπαίδευση στην GPU αποδείχθηκε 1.7 φορές ταχύτερη. Ωστόσο, αξίζει να σημειωθεί πως στον αλγόριθμο βελτιστοποίησης πολιτικής PPO, τα 1M βήματα εκπαίδευσης χρειάστηκαν 0.75h με χρήση CPU και 1.64h με χρήση GPU, δηλαδή η εκπαίδευση στην GPU αποδείχθηκε 2.2 φορές πιο αργή. Το αποτέλεσμα αυτό προκαλεί εντύπωση και πιθανόν να οφείλεται στη μειωμένη πολυπλοκότητα των υπολογισμών του αλγορίθμου PPO, σε σχέση με τους αλγορίθμους δράστη-κριτή. Με άλλα λόγια, μπορεί ο χρόνος αντιγραφής των δεδομένων από τη μνήμη του συστήματος στη GPU και αντίστροφα, να υπερτερεί του πλεονεκτήματος της υπολογιστικής ισχύος της, χάρη στην απλότητα των υπολογισμών του αλγορίθμου PPO.

## 5.4 Στατιστικά

Σε αυτήν την ενότητα, θα παρουσιάσουμε κάποια συνολικά στατιστικά της διαδικασίας εκπαίδευσης και βασιζόμενοι σε αυτά, θα βγάλουμε κάποια πρώτα συμπεράσματα.

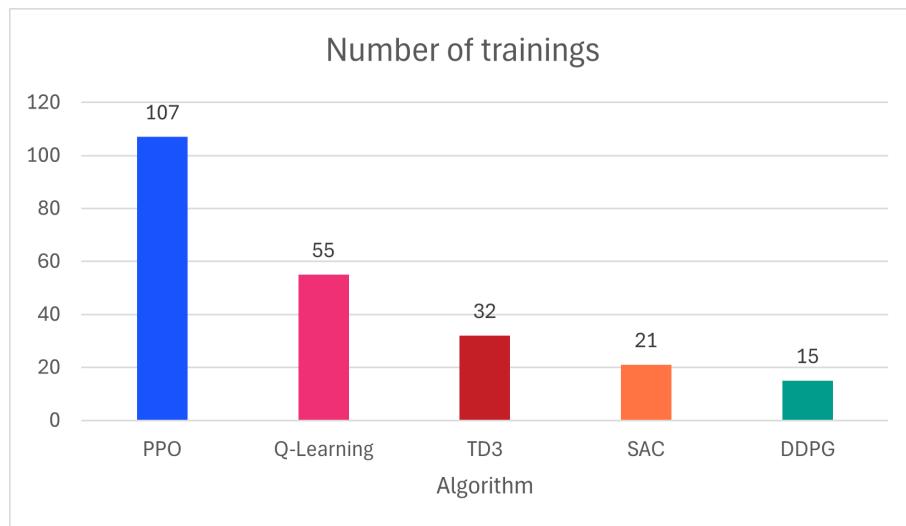
Αρχικά, η συνολική διάρκεια των εκπαιδεύσεων των αλγορίθμων, ανέρχεται σε περίπου 98 ημέρες και 8 ώρες<sup>2</sup>. Βέβαια, η διάρκεια αυτή δεν αντιστοιχεί σε πραγματικό χρόνο, αφού όπως εξηγήσαμε στην προηγούμενη ενότητα, οι εκπαιδεύσεις διεξάγονταν συχνά παράλληλα, σε διαφορετικά υπολογιστικά συστήματα. Είναι όμως ενδεικτική του χρόνου που απαιτείται για την εκπαίδευση τέτοιων αλγορίθμων.

Στην *Eικόνα 5.2* παρουσιάζεται το πλήθος των διαφορετικών εκπαιδεύσεων που διεξήχθησαν για κάθε αλγόριθμο.

Η εικόνα αυτή αποτελεί ένα πρώτο σημάδι της τελικής απόδοσης των αλγορίθμων. Συγκεκριμένα, διεξήχθησαν περισσότερες εκπαιδεύσεις με τους αλγορίθμους Q-Learning και PPO, καθώς ήταν αυτοί

<sup>1</sup>Το εργαλείο CUDA είναι μία πλατφόρμα παράλληλης υπολογιστικής αρχιτεκτονικής, που επιτρέπει τη χρήση των GPU της NVidia για γενικού σκοπού υπολογισμούς, επιταχύνοντας έτσι την εκτέλεση τους.

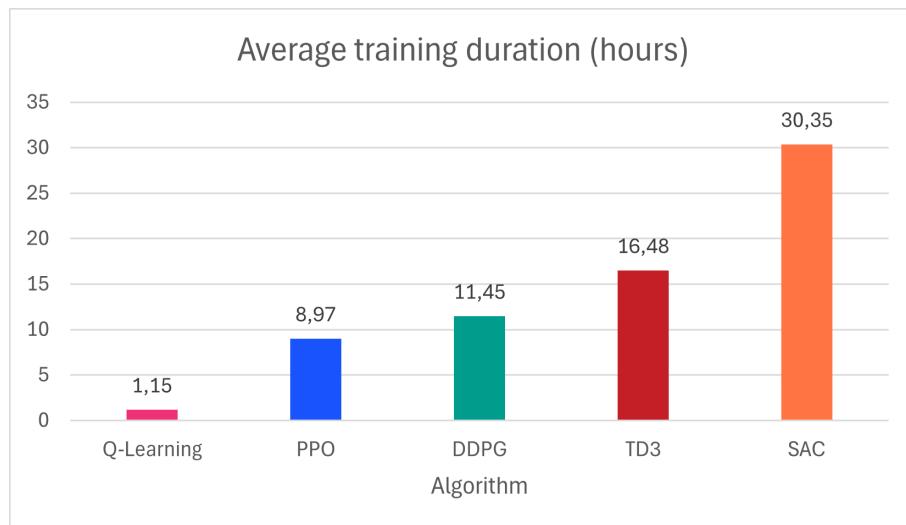
<sup>2</sup>Για τον υπολογισμό αυτό χρησιμοποιήθηκαν οι ώρες κάθε εκπαίδευσης, όπως φαίνονται από τα γραφήματα του Tensorboard και το εργαλείο [Time Calculator](#), για την πρόσθεση τους.



**Εικόνα 5.2.** Πλήθος εκπαιδεύσεων ανά αλγόριθμο.

με τις χειρότερες επιδόσεις, τις οποίες προσπαθούσαμε να βελτιώσουμε. Αντίθετα, οι αλγόριθμοι της κατηγορίας δράστη-κριτή δεν χρειάστηκαν τόσο μεγάλο πλήθος εκπαιδεύσεων, αφού είχαν πιο γρήγορα, καλές επιδόσεις.

Στη συνέχεια, στην Εικόνα 5.3 διακρίνεται η μέση διάρκεια εκπαίδευσης των αλγορίθμων, σε ώρες.



**Εικόνα 5.3.** Μέση διάρκεια εκπαίδευσης ανά αλγόριθμο.

Παρατηρούμε πως οι αλγόριθμοι που χρησιμοποιούν νευρωνικά δίκτυα, απαιτούν σημαντικά περισσότερο χρόνο για την εκπαίδευσή τους, σε σχέση με τον απλό αλγόριθμο Q-Learning. Ακόμα, στους αλγορίθμους βαθιάς ενισχυτικής μάθησης, ο αλγόριθμος βελτιστοποίησης πολιτικής PPO,

## 5.5 Δυσκολίες Ενισχυτικής Μάθησης

---

χρειάζεται λιγότερο χρόνο εκπαίδευσης, σε σχέση με αυτούς της κατηγορίας δράστη-κριτή. Επομένως, επιβεβαιώνουμε πως η πολυπλοκότητα του κάθε αλγορίθμου επηρεάζει τον χρόνο εκπαίδευσής του.

## 5.5 Δυσκολίες Ενισχυτικής Μάθησης

Σε αυτήν την ενότητα θα αναλύσουμε τους διάφορους παράγοντες που καθιστούν τη διαδικασία της εκπαίδευσης πρακτόρων ενισχυτικής μάθησης, απαιτητική και πολύπλοκη. Θα εξετάσουμε τους λόγους δυσκολίας σε θεωρητικό επίπεδο, όπως αυτοί περιγράφονται στη βιβλιογραφία, ενώ παράλληλα θα αναφερθούμε και στην εμφάνισή τους κατά τη διάρκεια των εκπαιδεύσεων που διεξήχθησαν.

### 5.5.1 Χρόνος εκπαίδευσης

Ίσως ο βασικότερος λόγος δυσκολίας της ενισχυτικής μάθησης, είναι η μη αποδοτικότητα των αλγορίθμων στη χρήση δειγμάτων από το περιβάλλον (*sample inefficient*), η οποία οδηγεί σε πολύ υψηλούς χρόνους εκπαίδευσης. Το άρθρο «Deep Reinforcement Learning Doesn’t Work Yet» (Igelnik 2018), το οποίο αναλύει τα κυριότερα προβλήματα της βαθιάς ενισχυτικής μάθησης, αναφέρει χαρακτηριστικά πως ο αλγόριθμος Rainbow DQN, ένας από τους πιο προηγμένους αλγορίθμους ενισχυτικής μάθησης, απαιτεί περίπου 83 ώρες εμπειρίας σε ένα απλό παιχνίδι Atari, προκειμένου να φτάσει τη μέση ανθρώπινη επίδοση. Ο χρόνος αυτός είναι πολύ μεγάλος, αν αναλογιστεί κανείς πως ένας άνθρωπος μπορεί να μάθει να παίζει το ίδιο παιχνίδι, σε μόλις μερικά λεπτά.

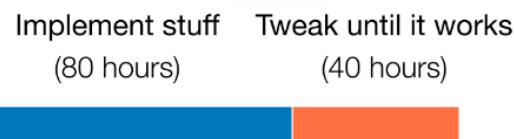
Πράγματι, το πρόβλημα των μεγάλων χρόνων εκπαίδευσης επιβεβαιώθηκε στην εργασία μας, όπου συνήθως χρειαζόταν ένα χρονικό διάστημα περίπου ίσο με μία ημέρα, προκείμενου να μπορούμε να κρίνουμε με ασφάλεια την πορεία της εκπαίδευσης. Αυτό μετατρέπει τη διαδικασία της εκπαίδευσης σε έναν μαραθώνιο, όπου οι αλλαγές που γίνονται, απαιτούν πολύ χρόνο για να αξιολογηθούν. Από την άλλη, οι σχεδιαστές του συστήματος, προσπαθούμε να μην σπαταλάμε χρόνο σε ανώφελες εκπαιδεύσεις, γιατί γνωρίζουμε πως ο χρόνος που έχουμε στη διάθεση μας πρέπει να χρησιμοποιηθεί αποδοτικά. Έτσι, ο μεγάλος χρόνος εκπαίδευσης, μας κάνει συχνά διστακτικούς απέναντι στον πειραματισμό, ωθώντας μας σε πιο συντηρητικές και γνώριμες επιλογές, για τα αποτελέσματα των οποίων είμαστε πιο σίγουροι.

### 5.5.2 Αποσφαλμάτωση

Η ενισχυτική μάθηση είναι δύσκολη στην αποσφαλμάτωση (*debugging*) και στην ερμηνεία των αποτελεσμάτων της. Πιο αναλυτικά, δεν είναι πάντα σαφές γιατί ο πράκτορας συμπεριφέρεται με συγκεκριμένο τρόπο, καθώς υπάρχει πληθώρα παραγόντων που μπορεί να δυσχεραίνει την απόδοση

του. Για παράδειγμα, ο σχεδιαστής του συστήματος πρέπει να αναλογιστεί την αρχιτεκτονική των νευρωνικών δικτύων, τις υπερ-παραμέτρους του αλγορίθμου, την ποιότητα της συνάρτησης ανταμοιβής ή ακόμα και της κατάστασης που δέχεται ως είσοδο ο πράκτορας, καθώς και των διαθέσμων ενεργειών του. Επομένως, καθίσταται δύσκολη η σωστή διάγνωση των προβλημάτων της εκπαίδευσης και η επίλυσή τους. Η χρονοβόρα διαδικασία της αποσφαλμάτωσης και των τροποποιήσεων της εκπαίδευσης αναφέρεται και στο (Amid 2018). Μάλιστα, χαρακτηριστική είναι η *Εικόνα 5.4*, όπου ο συγγραφέας παρουσιάζει τη διαφορά μεταξύ των εκτιμήσεων του, για την κατανομή του χρόνου σε ένα έργο ενισχυτικής μάθησης και της πραγματικότητας.

### Expectation:



### Reality:



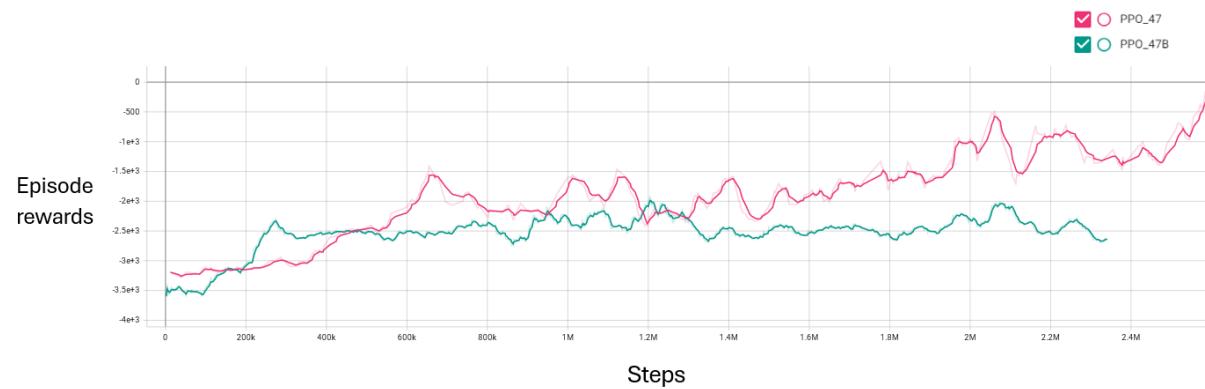
**Εικόνα 5.4.** Προσδοκίες και πραγματικότητα στην εκπαίδευση πρακτόρων ενισχυτικής μάθησης (Amid 2018).

Παρατηρούμε πως ο χρόνος της διόρθωσης παραμέτρων της εκπαίδευσης αποδείχθηκε πολύ μεγαλύτερος από τον αναμενόμενο. Αυτό εναρμονίζεται και με τη δική μας εμπειρία. Παρόλο που δεν καταγράψαμε ακριβώς τον χρόνο υλοποίησης του κώδικα, αυτός ήταν σίγουρα, σημαντικά μικρότερος του χρόνου που διήρκησαν οι εκπαιδεύσεις και οι τροποποιήσεις τους, προκειμένου να επιτευχθούν ικανοποιητικά αποτελέσματα.

### 5.5.3 Υπερ-παραμέτροι

Όπως περιγράφεται και στο (Patel, Carver, και Rahimi 2011), η ενισχυτική μάθηση απαιτεί την επιλογή τιμών για πολλές υπερ-παραμέτρους, οι οποίες μπορεί να επηρεάσουν σημαντικά την απόδοση του αλγορίθμου. Ωστόσο, η βιβλιογραφία παρέχει περιορισμένες κατευθυντήριες οδηγίες για την ρύθμιση αυτών των παραμέτρων. Η επιλογή των σωστών τιμών για τις υπερ-παραμέτρους, μπορεί να χρειαστεί πολλές δοκιμές, καθιστώντας τη διαδικασία εκπαίδευσης χρονοβόρα και απαιτητική.

Η ευαισθησία των αλγορίθμων στις τιμές των παραμέτρων εντοπίστηκε και σε αυτήν την εργασία. Για παράδειγμα, στην *Εικόνα 5.5*, παρουσιάζονται δύο διαφορετικές εκπαίδευσεις του ίδιου αλγορίθμου, όπου η μονή διαφορά είναι μία μικρή μεταβολή της τιμής του συντελεστή εντροπίας.



**Εικόνα 5.5.** Δύο εκπαίδευσεις του αλγορίθμου PPO: η εκπαίδευση 47 με entropy coefficient = 0.01 και η εκπαίδευση 47B με entropy coefficient = 0.

Παρατηρούμε πως η μικρή αυτή μεταβολή στην τιμή της υπερ-παραμέτρου, οδήγησε πράγματι, σε άλλαγη της απόδοσης του αλγορίθμου, αφού ήδη από τα 1.4M steps, η γραφική των episode rewards της εκπαίδευσης 47B συγκλίνει σε αρνητική τιμή, ενώ αυτή της εκπαίδευσης 47 συνεχίζει να αυξάνεται. Ωστόσο, πρέπει να σημειωθεί, πως οι βελτιώσεις της απόδοσης που πετύχαμε κάνοντας τέτοιου είδους μεταβολές ήταν πάντοτε μικρές και δεν αποτέλεσαν την αιτία για την επίτευξη της επιθυμητής συμπεριφοράς του πράκτορα. Με άλλα λόγια, επιβεβαιώσαμε την επιρροή των υπερ-παραμέτρων στην εκπαίδευση, όμως αυτές ποτέ δεν ήταν η ειδοποιός διαφορά μεταξύ μίας αποτυχημένης και μίας επιτυχημένης εκπαίδευσης.

### 5.5.4 Σχεδίαση συνάρτησης ανταμοιβής

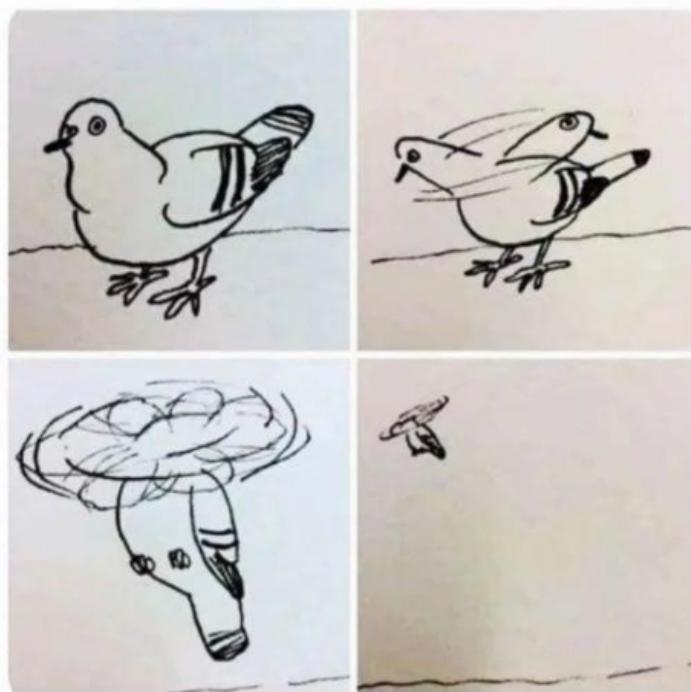
Η ενισχυτική μάθηση εξαρτάται σε μεγάλο βαθμό, από την ποιότητα της συνάρτησης ανταμοιβής. Αν αυτή δεν είναι άρτια σχεδιασμένη, ο πράκτορας μπορεί να μην μάθει τη βέλτιστη πολιτική. Η σωστή σχεδίαση είναι πιο δύσκολη από ότι φαίνεται, καθώς η συνάρτηση ανταμοιβής πρέπει να αντικατοπτρίζει με ακρίβεια τον πραγματικό στόχο του πράκτορα και να τον ωθεί αποκλειστικά

στην επιθυμητή συμπεριφορά. Συχνά, οι αστοχίες στην εκπαίδευση οφείλονται στην παρερμηνεία της συνάρτησης ανταμοιβής από τον πράκτορα. Αυτό είναι ένα πλεονέκτημα των έτοιμων περιβαλλοντών εκπαίδευσης (π.χ. Atari), που τα καθιστά δημοφιλή, καθώς η συνάρτηση ανταμοιβής είναι απλώς το σκορ του παιχνιδιού και δεν χρειάζεται να οριστεί από τον σχεδιαστή του συστήματος.

### Πειρατεία της ανταμοιβής

Ένα φαινόμενο που μπορεί να προκύψει από την ελαττωματική σχεδίαση της συνάρτησης ανταμοιβής, είναι η **πειρατεία της ανταμοιβής** (*reward hacking*). Ο όρος αυτός, περιγράφει έναν έξυπνο, αντισυμβατικό τρόπο που βρήκε ο πράκτορας για να μεγιστοποιεί την ανταμοιβή του, χωρίς επιτυγχάνει τον πραγματικό στόχο που είχαν κατά νου οι σχεδιαστές του (Miles 2024). Επομένως, ο πράκτορας ανακαλύπτει κενά στο περιβάλλον του ή εκμεταλλέυεται αστοχίες του λογισμικού, με αποτέλεσμα να πετυχαίνει μεγαλύτερη συνολική ανταμοιβή, από αυτήν που θα πετυχαίνει ακολουθώντας την επιθυμητή συμπεριφορά. Ένα παράδειγμα reward hacking, παρουσιάζεται με χιουμοριστικό τρόπο, στην *Εικόνα 5.6*.

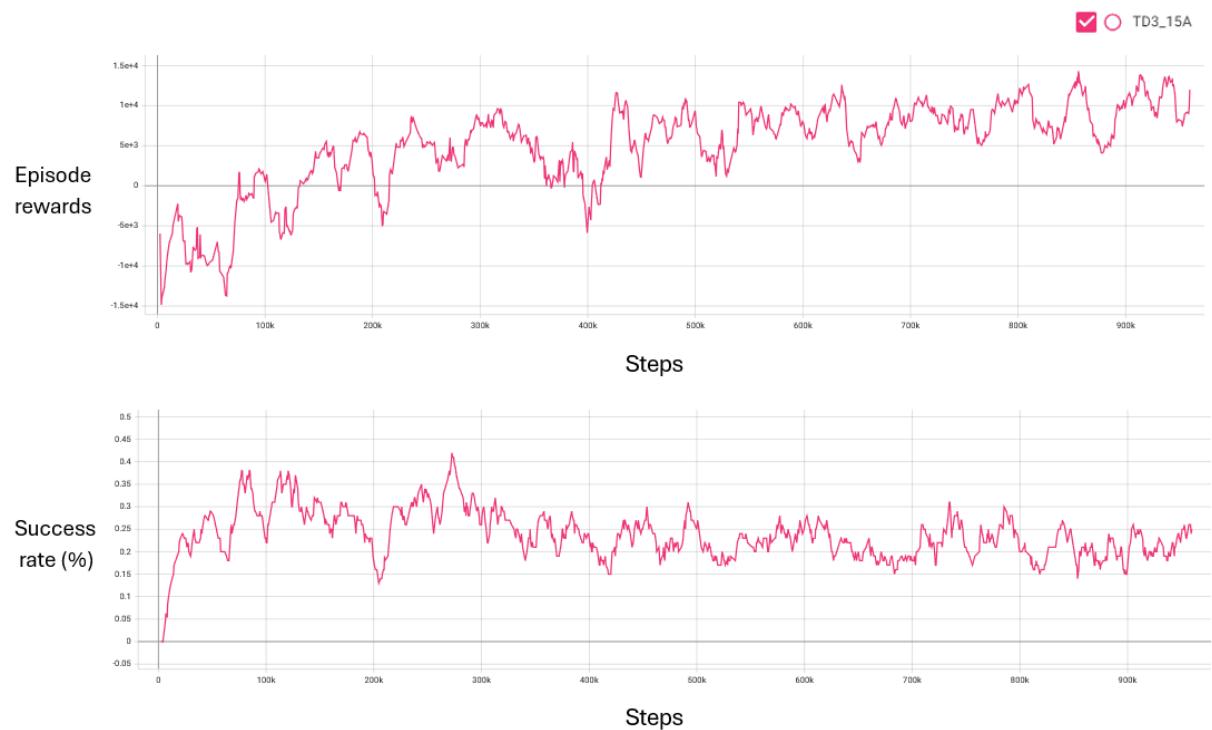
## When your RL agent learns to glitch the environment



**Εικόνα 5.6.** Παράδειγμα reward hacking (Raffin 2021).

## 5.5 Δυσκολίες Ενισχυτικής Μάθησης

Μία ενδιαφέρουσα περίπτωση reward hacking προέκυψε στις εκπαιδεύσεις μας, όταν προσπαθούσαμε μέσω της διαμόρφωσης της ανταμοιβής (*reward shaping*, βλ. 3.3.2), να ωθήσουμε τον πράκτορα να παρκάρει, δηλαδή να παραμείνει ακίνητος εντός της θέσης στάθμευσης. Για να το πετύχουμε αυτό, στη συνάρτηση ανταμοιβής επιβραβεύαμε τον πράκτορα, σε κάθε βήμα που βρισκόταν εντός της θέσης στάθμευσης (+100). Μάλιστα, η επιβράβευση αυτή, ήταν αντιστρόφως ανάλογη της ταχύτητας του πράκτορα, ώστε να τον ενθαρρύνουμε να μην κινείται. Τέλος, όταν ο πράκτορας παρέμενε για 2 συνεχόμενα δευτερόλεπτα ακίνητος εντός της θέσης, τότε θεωρούσαμε πως πάρκαρε επιτυχώς και του δίναμε μεγάλη επιβράβευση (+5000). Ωστόσο, από τις γραφικές παραστάσεις του Tensorboard, οι οποίες φαίνονται στην Εικόνα 5.7, παρατηρήσαμε το παράδοξο, πως η μέση ανταμοιβή του πράκτορα αυξανόταν πολύ πέραν της ανταμοιβής για τη στάθμευση του, ενώ το ποσοστό επιτυχίας του παρέμενε μικρό (~25%).



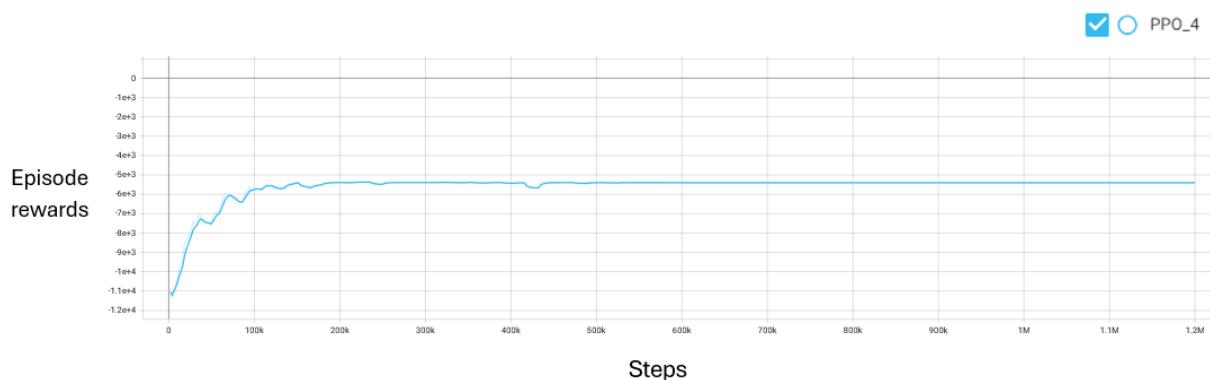
**Εικόνα 5.7.** Περίπτωση reward hacking στο περιβάλλον αυτόματης στάθμευσης. Ο πράκτορας έμαθε να πετυχαίνει μεγαλύτερη ανταμοιβή, χωρίς να παρκάρει.

Εξετάζοντας τον πράκτορα, παρατηρήσαμε πως είχε μάθει να εισέρχεται αρχικά στη θέση στάθμευσης, αλλά στη συνέχεια να κινείται διαδοχικά εμπρός-πίσω, ώστε να λαμβάνει συνεχώς τις μικρές επιβραβεύσεις που του δίναμε, όσο βρίσκεται εντός της θέσης. Έτσι, δεν τον ενδιέφερε να παρκάρει, καθώς οι συνολικές ανταμοιβές που λάμβανε από αυτή τη συμπεριφορά (της τάξης των 10000), ήταν πολύ μεγαλύτερες, από αυτές του παρκαρίσματος (5000). Η λύση σε αυτό το πρόβλημα, δόθηκε από την πιο προσεκτική ρύθμιση των τιμών των επιβραβεύσεων, σε κάθε περίπτωση.

#### 5.5.4.1 Υιοθέτηση υποβέλτιστης πολιτικής

Παρόλο που το reward hacking αποτελεί ένα από τα πιο ενδιαφέροντα φαινόμενα της ενισχυτικής μάθησης, τέτοιες περιπτώσεις είναι αρκετά σπάνιες. Το πολύ συνηθέστερο πρόβλημα, που αντιμετωπίζουν οι πράκτορες ενισχυτικής μάθησης, είναι η σύγκλιση σε τοπικό μέγιστο της συνάρτησης ανταμοιβής, δηλαδή την υιοθέτηση υποβέλτιστης πολιτικής από τον πράκτορα. Αυτό σημαίνει πως ο πράκτορας έχει μάθει κάποια χρήσψη συμπεριφορά, η οποία ενισχύει τη συνολική ανταμοιβή του. Ωστόσο, η συμπεριφορά αυτή, είναι πολύ μακρία από την επιθυμητή από τους σχεδιαστές του συστήματος.

Από την εμπειρία μας, αυτό ήταν το πιο συχνό πρόβλημα που συναντήσαμε, στις διάφορες εκπαιδεύσεις πρακτόρων. Ένα παράδειγμα υιοθέτησης υποβέλτιστης πολιτικής, το οποίο συνέβη πολλές φορές και σε διαφορετικούς αλγορίθμους, περιγράφεται παρακάτω. Αρχικά, στη συνάρτηση ανταμοιβής, προσθέτουμε μία μικρή τιμωρία στον πράκτορα για κάθε βήμα που περνάει (-9), με το σκεπτικό να τον αθήσουμε να παρκάρει το συντομότερο δυνατόν. Επίσης, προσθέτουμε μία μεγαλύτερη τιμωρία (-100), η οποία δίνεται στον πράκτορα, όταν αυτός συγκρούεται με άλλα αντικείμενα. Ακόμα, υπάρχουν οι κατάλληλες επιβραβέυσεις, για όταν ο πράκτορας πλησιάζει τη θέση στάθμευσης, όταν εισέλθει σε αυτήν και όταν παρκάρει. Ωστόσο, η γραφική παράσταση της μέσης ανταμοιβής του πράκτορα για αυτήν την εκπαίδευση, συγκλίνει γρήγορα στην τιμή -5400, όπως φαίνεται στην Εικόνα 5.8.



Εικόνα 5.8. Παράδειγμα σύγκλισης σε τοπικό μέγιστο.

Εξετάζοντας στην πράξη τον πράκτορα, παρατηρήσαμε πως είχε μάθει απλώς να μένει ακίνητος σε κάθε επεισόδιο, για όλη τη διάρκεια του (εξού και  $-9 \times 600 \text{ steps} = -5400$  average episode reward). Αυτό που προφανώς συνέβη, ήταν πως στην αρχή της εκπαίδευσης, όπου οι ενέργειες του πράκτορα είναι τυχαίες, αυτός αναπόφευκτα, είχε συχνές συγκρούσεις με άλλα αντικείμενα. Έτσι, η συνολική ανταμοιβή του ήταν πολύ αρνητική και ο πράκτορας έμαθε πως η καλύτερη στρατηγική, για να μεγιστοποιήσει την ανταμοιβή του, είναι να μένει ακίνητος. Πράγματι, όπως βλέπουμε από την Εικόνα 5.8, ο πράκτορας βέλτιωσε σημαντικά τη συνολική ανταμοιβή του, όμως αυτή είναι πολύ μικρότερη

## 5.5 Δυσκολίες Ενισχυτικής Μάθησης

---

από την ανταμοιβή που θα έπαιρνε, αν πάρκαρε. Ωστόσο, ο πράκτορας πλέον, έχει συγκλίνει σε αυτήν την πολιτική και δεν εξερευνεί παραπάνω το περιβάλλον.

Επιχειρήσαμε να αντιμετώπισουμε αυτό το πρόβλημα, προσθέτωντας μία τιμωρία στην ανταμοιβή του πράκτορα, για κάθε βήμα που παρέμενε ακίνητος, ώστε να τον ενθαρρύνουμε να εξερευνήσει περισσότερο και να βρει τη βέλτιστη πολιτική. Παρόλα αυτά, η απάντηση του πράκτορα ήταν να μάθει να κάνει τον γύρο της πίστας, ασταμάτητα. Με τον τρόπο αυτό, απέφευγε τόσο την τιμωρία για τις συγκρούσεις, όσο και την τιμωρία για την ακινησία, αλλά και πάλι, δεν ανέπτυξε την επιθυμητή συμπεριφορά. Επομένως, γίνεται κατανοητό πως το μέγεθος της κάθε ανταμοιβής πρέπει να επιλεγεί πολύ προσεκτικά, ώστε να μην παρακινήσει τον πράκτορα σε κάποια υποβέλτιστη συμπεριφορά.

Η τάση αυτή των πρακτόρων ενισχυτικής μάθησης να συγκλίνουν σε υποβέλτιστες πολιτικές, αντί για τις επιθυμητές από τους σχεδιαστές τους, ώθησε τον (Irpan 2018), να τους παρομοιάσει με «τεμπέλικους δαίμονες, που προσπαθούν εσκεμμένα να παρερμηνεύσουν την ανταμοιβή και αναζητούν ενεργά το πιο εύκολο τοπικό μέγιστο». Μία παρόμοια σκέψη κάναμε και εμείς, βλέποντας συμπεριφορές όπως αυτή που περιγράψαμε πριν. Θεωρήσαμε δηλαδή, πως είναι παραγωγικό να φανταζόμαστε τους πράκτορες σαν μικρά παιδιά, τα οποία προσπαθούν επίτηδες, να αντιτίθενται στη συμπεριφορά που τους ζητάμε.

Προφανώς, αυτές οι θεωρήσεις προέρχονται από την απογοήτευση των σχεδιαστών και έχουν κυρίως, ψυχαγωγική διάθεση. Το πραγματικό πρόβλημα, έγκειται στη δυσκολία εύρεσης ισορροπίας στο δίλημμα εξερεύνησης-εκμετάλλευσης. Συγκεκριμένα, όταν ο πράκτορας εξερευνεί πολύ, τα δείγματα που συλλέγει είναι ανεπαρκή για να μάθει τη βέλτιστη συμπεριφορά. Αντίθετα, όταν εκμεταλλεύεται πολύ, κινδυνεύει να συγκλίνει πρόωρα, σε συμπεριφορές που δεν είναι βέλτιστες.

Δυστυχώς, παρόλο που το πρόβλημα της εξερεύνησης είναι από τα παλαιότερα του πεδίου της ενισχυτικής μάθησης και έχουν προταθεί διάφορες ιδέες κατά καιρούς για την αντιμετώπιση του, καμία δεν εγγυάται σταθερά αποτελέσματα σε όλα τα περιβάλλοντα. Ο λόγος για αυτό είναι το μέγεθος της δυσκολίας του προβλήματος. Σύμφωνα με τη Wikipedia, το πρόβλημα εντοπίστηκε αρχικά από ερευνητές των συμμάχων στον 2<sup>ο</sup> παγκόσμιο πόλεμο, όμως αποδείχθηκε τόσο δυσεπίλυτο, που προτάθηκε η διαρροή του σε Γερμανούς επιστήμονες, ώστε να σπαταλήσουν και αυτοί τον χρόνο τους πάνω σε αυτό (Wikipedia 2024).

### 5.5.5 Αστάθεια της εκπαίδευσης

Ένας παράγοντας που προκαλεί αστάθεια στις εκπαιδεύσεις ενισχυτικής μάθησης, είναι το γεγονός πως τα δεδομένα συλλέγονται από τον ίδιο τον πράκτορα, κατά την αλληλεπίδραση με το περιβάλλον του. Αυτό έρχεται σε αντίθεση με την πολύ πιο σταθερή, επιβλεπόμενη μάθηση, όπου υπάρχει ένα στατικό σύνολο δεδομένων, καθορισμένο πριν την εκπαίδευση. Έτσι, στην ενισχυτική μάθηση, τα δεδομένα εκπαίδευσης εξαρτώνται από την πολιτική του πράκτορα. Αυτή η εξάρτηση, μπορεί να

οδηγήσει σε φαύλο κύκλο: αν ο πράκτορας συλλέγει κακής ποιότητας δεδομένα (π.χ. καταστάσεις χωρίς ανταμοιβές), τότε δεν θα ανακαλύψει κάποια χρήσιμη συμπεριφορά και θα συνεχίσει να συλλέγει κακής ποιότητας δεδομένα κοκ.

Επομένως, προκύπτει πως δύο εκπαιδεύσεις ενισχυτικής μάθησης, με τα ίδια ακριβώς χαρακτηριστικά, μπορεί να έχουν διαφορετικά αποτελέσματα η μία με την άλλη. Συγκεκριμένα, σε μία εκπαίδευση όπου τυχαία, ανακαλύπτονται νωρίς καλά δείγματα, η απόδοση του πράκτορα θα βελτιωθεί. Αντίθετα, σε μία εκπαίδευση όπου δεν ανακαλύπτονται χρήσιμα δείγματα από νωρίς, ο πράκτορας μπορεί να συγκλίνει σε υποβέλτιστες λύσεις, καθώς βλέπει πως όσα δοκιμάζει, έχουν χειρότερα αποτελέσματα.

Άρα, πέραν των υπολοιπών προβλημάτων των εκπαιδεύσεων ενισχυτικής μάθησης, αυτές παρουσιάζουν και εξάρτηση από την τύχη, δηλαδή από τους ψευδο-τυχαίους αριθμούς που παράγονται από το περιβάλλον. Για αυτό, προτείνεται πάντα να εκτελούνται πολλαπλές εκπαιδεύσεις με τα ίδια χαρακτηριστικά, ώστε τα αποτελέσματα να θεωρούνται αξιόπιστα (Leger και Raffin 2024).

## 5.6 Καλές πρακτικές

Σε αυτήν την ενότητα, παρουσιάζονται κάποιες καλές πρακτικές στις εκπαιδεύσεις ενισχυτικής μάθησης, οι οποίες αξιοποιήθηκαν στην παρούσα εργασία και αποδείχθηκαν χρήσιμες για τη βελτίωση της απόδοσης των πρακτόρων. Όπως και πριν, θα εξετάσουμε τις συμβουλές αυτές πρώτα σε θεωρητικό επίπεδο, όπως περιγράφονται στη βιβλιογραφία, ενώ στη συνέχεια θα δούμε την εφαρμογή τους, στο περιβάλλον αυτόματης στάθμευσης.

### 5.6.1 Επιλογή αλγορίθμου

Στη βιβλιογραφία αναφέρεται πως δεν υπάρχει συγκεκριμένος κανόνας για το ποιος αλγόριθμος είναι καλύτερος, καθώς η απόδοση τους εξαρτάται από πολλούς παράγοντες. Επομένως, δεν υπάρχει κάποια φόρμουλα για την επιλογή αλγορίθμου, αλλά πρέπει να ληφθεί υπόψη το είδος του προβλήματος. Σημαντική είναι η διάκριση των χώρων καταστάσεων και ενεργειών του περιβάλλοντος, σε διακριτούς ή συνεχείς. Πιο αναλυτικά, υπάρχουν αλγόριθμοι που είναι ανεξάρτητοι αυτών των παραγόντων, όμως υπάρχουν κάποιοι που είναι κατάλληλοι μόνο για συγκεκριμένες περιπτώσεις. Οι χώροι για τους οποίους είναι κατάλληλοι, οι αλγόριθμοι που χρησιμοποιήθηκαν στην παρούσα εργασία, παρουσιάζονται στον Πίνακα 5.2:

## 5.6 Καλές πρακτικές

---

**Πίνακας 5.2:** Πίνακας καταλληλότητας αλγορίθμων ενισχυτικής μάθησης

Αλγόριθμος	Χώρος Καταστάσεων (State Space)	Χώρος Ενεργειών (Action Space)
Q-Learning	Διακριτός	Διακριτός
PPO	Συνεχής	Διακριτός ή Συνεχής
SAC	Συνεχής	Συνεχής
DDPG	Συνεχής	Συνεχής
TD3	Συνεχής	Συνεχής

Όπως φαίνεται από τον παραπάνω πίνακα, η επιλογή αλγορίθμου ήταν ένα πρώτο λάθος που κάναμε, ως αρχάριοι, σε αυτήν την εργασία. Συγκεκριμένα, η επιλογή του αλγορίθμου Q-Learning για την επίλυση του προβλήματος αυτόματης στάθμευσης ήταν άστοχη, καθώς ο αλγόριθμος δεν ενδείκνυται για περιβάλλοντα με συνεχή χώρο καταστάσεων, όπως το δικό μας. Έτσι, οι όποιες προσπάθειες έγιναν για την αντιμετώπιση αυτού του προβλήματος, όπως για παράδειγμα η διακριτοποίηση των καταστάσεων, δεν έφεραν τα επιθυμητά αποτελέσματα και σπαταλήθηκε άσκοπα, πολύτιμος χρόνος εκπαιδεύσεων. Επομένως, θα συμβούλευα τους νέους ερευνητές του πεδίου, να εξετάζουν προσεκτικά το state και το action space ενός αλγορίθμου, προτού προχωρήσουν σε εκπαιδεύσεις με αυτόν.

### 5.6.2 Επιλογή πολιτικής αλγορίθμου

Κάποιοι αλγόριθμοι ενισχυτικής μάθησης, όπως ο DDPG και ο TD3, χρησιμοποιούν ντετερμινιστική πολιτική, ενώ άλλοι, όπως ο PPO και ο SAC, χρησιμοποιούν στοχαστική πολιτική. Η στοχαστική πολιτική, μπορεί να είναι χρήσιμη κατά την εκπαίδευση, ενθαρρύνοντας την εξερεύνηση του περιβάλλοντος, όμως, κατά την αξιολόγηση του πράκτορα, είναι καλό να θέτουμε πάντα την πολιτική του πράκτορα σε ντετερμινιστική. Με αυτόν τον τρόπο, είμαστε σε θέση να κρίνουμε καλύτερα τη συμπεριφορά που έχει μάθει ο πράκτορας, ενώ συνήθως βελτιώνονται και οι επιδόσεις του, αφού επιλέγονται πάντα οι ενέργειες, που έκρινε ως βέλτιστες ο αλγόριθμος.

### 5.6.3 Μοντελοποίηση προβλήματος

#### Χώροι καταστάσεων και ενεργειών

Δύο σημαντικές παράμετροι της μοντελοποίησης του προβλήματος είναι η επιλογή της κατάστασης του περιβάλλοντος (είσοδος του πράκτορα) και των δυνατών ενεργειών του πράκτορα (εξοδος του πράκτορα). Συγκεκριμένα, είναι σημαντικό η κατάσταση του περιβάλλοντος να περιέχει μόνο

χρήσιμες πληροφορίες για τον πράκτορα, προκειμένου να ελαχιστοποιηθεί το πλήθος των εισόδων του νευρωνικού δικτύου. Ακόμα, πρέπει ο πράκτορας να μπορεί να συσχετίσει τις ανταμοιβές που παίρνει, σε συγκεκριμένες καταστάσεις. Για παράδειγμα, η τιμωρία της σύγκρουσης μπορεί να γίνει κατανόητη από τον πράκτορα, καθώς παρατηρεί ότι τιμωρείται όταν η απόσταση ενός αισθητήρα γίνει 0. Μάλιστα, παρέχοντας ως είσοδο στον πράκτορα και την ταχύτητα του, θα είναι σε θέση να καταλάβει πως όταν η τιμή του αισθητήρα είναι μικρή και η ταχύτητα του μεγάλη, έπεται σύγκρουση.

Αντίστοιχα, οι ενέργειες του πράκτορα πρέπει να επαρκούν για την επίλυση του προβλήματος. Μία σημαντική επιλογή που πρέπει να γίνει, είναι αυτή μεταξύ διακριτού ή συνεχούς χώρου ενεργειών. Ο διακριτός χρόνος ενεργειών θεωρείται πως προσφέρει ταχύτερη και ευκολότερη εκπαίδευση, ενώ ο συνεχής, καλύτερη τελική απόδοση. Στην περίπτωση μας, επιλέχθηκε διακριτός χώρος ενεργειών, ώστε ο πράκτορας να χειρίζεται το αυτοκίνητο με τα βέλη του πληκτρολογίου, όπως θα έκανε ένας άνθρωπος για να παίξει το παιχνίδι. Έτσι, ακόμα και όταν χρησιμοποιήθηκαν αλγόριθμοι με συνεχή χώρο ενεργειών, οι ενέργειες τους μετατράπηκαν στη συνέχεια σε διακριτές, ώστε να αντιστοιχούν στα βέλη του πληκτρολογίου. Με τον τρόπο αυτό, θεωρήσαμε πως μπορούμε να αξιολογήσουμε τους αλγορίθμους σε ίσους όρους, καθώς και να τους συγκρίνουμε με την ανθρώπινη απόδοση στο παιχνίδι. Μάλιστα, αυτό επιβεβαιώνεται, καθώς όταν υπό αυτές τις συνθήκες, συγκρίθηκε η χρήση διακριτών και συνεχών ενεργειών στον αλγόριθμο PPO, δεν παρατηρήθηκε καμία διαφορά στην απόδοση του πράκτορα.

### **Αραιές ανταμοιβές έναντι Διαμόρφωσης ανταμοιβής**

Από την εμπειρία μας, η συνάρτηση ανταμοιβής ήταν ο παράγοντας που επηρέαζε στο μεγαλύτερο βαθμό, την απόδοση των πρακτόρων. Για το πρόβλημα της αυτόματης στάθμευσης, δοκιμάστηκαν διαφορετικές συναρτήσεις ανταμοιβής με πολλούς συνδυασμούς τιμών των ανταμοιβών τους και τα συμπεράσματα που προέκυψαν περιγράφονται παρακάτω.

Στη βιβλιογραφία, αναφέρεται πως μέσω αραιών ανταμοιβών, η εκπαίδευση του πράκτορα γίνεται συχνά πιο δύσκολη, εξαιτίας της έλλειψης επαρκούς ανάδρασης. Με άλλα λόγια, ο πράκτορας δεν έχει τρόπο να καταλάβει πόσο κοντά βρίσκεται στον τελικό στόχο του και σε κάποια περιβάλλοντα, είναι δύσκολο να επιλέξει τη σωστή αλληλουχία ενεργειών, που θα τον οδηγήσει στο στόχο και στην ανταμοιβή του, χωρίς καμία παρότρυνση από τον σχεδιαστή. Για αυτό, η διαμόρφωση ανταμοιβής θεωρείται πιο εύκολη για τη μάθηση του πράκτορα και προτείνεται οι εκπαίδευσεις να ξεκινάνε με αυτή τη μέθοδο (Raffin 2021).

Παρόλα αυτά, η διαμόρφωση ανταμοιβής διαθέτει τις δικές της προκλήσεις. Συγκεκριμένα, οι τιμές των ανταμοιβών πρέπει να οριστούν πολύ προσεκτικά, ώστε να μην εμφανίζεται το φαινόμενο του reward hacking. Αντίστοιχα, μέσω των τιμών των ανταμοιβών πρέπει να γίνεται σαφής διαχωρισμός μεταξύ πρωτεύοντος στόχου (π.χ. στάθμευση) και δευτερευόντων στόχων (π.χ. αποφυγή σύγκρουσης), προκειμένου ο πράκτορας να επικεντρωθεί στον κύριο στόχο του και να μην αναπτύξει υποβέλτιστες

πολιτικές. Γενικά, είναι σημαντικό ο σχεδιαστής να σκέπτεται από τη μεριά του πράκτορα, ο οποίος δεν γνωρίζει ποιός είναι ο στόχος του και προσπαθεί απλώς να μεγιστοποιήσει την ανταμοιβή του.

Τα παραπάνω προβλήματα της διαμόρφωσης ανταμοιβής είναι γνωστά στη βιβλιογραφία και αποδείχθηκαν στην περίπτωση μας, δύσκολο να ξεπεραστούν. Ακόμα, η μέθοδος των αραιών ανταμοιβών στάθηκε ικανή να επιτύχει την επιθυμητή απόδοση των πρακτόρων, σε ορισμένους αλγορίθμους. Για αυτούς τους λόγους, θα προέτρεπα μελλοντικούς ερευνητές να ξεκινήσουν από την σαφώς ευκολότερη μέθοδο των αραιών ανταμοιβών και εφόσον αυτή αποτύχει, να προχωρήσουν στη διαμόρφωση ανταμοιβής.

### Συνθήκες τερματισμού επεισοδίου

Είναι συνηθισμένη πρακτική να ορίζεται ένας μέγιστος αριθμός βημάτων, ως συνθήκη τερματισμού του επεισοδίου και έναρξης του επόμενου, κατά την εκπαίδευση ενός πράκτορα ενισχυτικής μάθησης. Η τεχνική αυτή ονομάζεται *episode cutoffs* ή *timeouts*. Με τον τρόπο αυτό, αποφέυγονται τα ατέρμονα επεισόδια, στα οποία ο πράκτορας έχει κολλήσει σε μία μη τερματική κατάσταση και δεν μπορεί να προχωρήσει περαιτέρω. Επιπλέον, η μέθοδος αυτή, ωθεί τον πράκτορα να ανακαλύπτει πιο αποδοτικές πολιτικές, καθώς πρέπει να επιλύσει το πρόβλημα εντός συγκεκριμένου αριθμού βημάτων. Επομένως, ενθαρρύνεται η ταχύτητα και η αποτελεσματικότητα του πράκτορα.

Στην περίπτωση μας, επιλέξαμε το άνω όριο των 600 βημάτων για τα επεισόδια της εκπαίδευσης. Αυτό το όριο επιλέχθηκε, καθώς θεωρήθηκε πως 30 δευτερόλεπτα αρκούν για να λυθεί το πρόβλημα της αυτόματης στάθμευσης, κάτω από οποιεσδήποτε αρχικές συνθήκες (30 seconds  $\times$  20 frames per second = 600 frames ή steps). Ταυτόχρονα, τα 30 δευτερόλεπτα είναι ένα σχετικά μικρό χρονικό διάστημα, έτσι ώστε να ασκείται πίεση στον πράκτορα, να επιτύχει τον στόχο του, εντός αυτού του χρόνου. Βέβαια, αξίζει να σημειωθεί, πως όταν δοκιμάσαμε να αφαιρέσουμε το όριο των 600 βημάτων, δεν παρατηρήσαμε μεγάλη διαφορά στα αποτελέσματα της εκπαίδευσης.

Μία άλλη τεχνική που εξετάστηκε, είναι αυτή του πρόωρου τερματισμού (*early stopping*). Η τεχνική αυτή, εισάγει επιπλέον συνθήκες για τον τερματισμό ενός επεισοδίου, όταν η απόδοση του πράκτορα σε αυτό, είναι πολύ μικρή. Για παράδειγμα, εφαρμόσαμε *early stopping* στα επεισόδια εκπαίδευσης, στην 3<sup>η</sup> σύγκρουση του πράκτορα με κάποιο αντικείμενο. Το σκεπτικό πίσω από αυτήν την τεχνική, είναι πως αποτρέπει την εξερεύνηση άχρηστων συμπεριφορών από τον πράκτορα (όπως το να συγκρούεται συνεχώς με αντικείμενα) και με αυτόν τον τρόπο, μπορεί να επιταχύνει την εκπαίδευση. Χρειάζεται όμως ξανά, προσεκτικός σχεδιασμός, καθώς μπορεί να προκύψει κι έτσι, reward hacking. Στο προηγούμενο παράδειγμα, αν ο πράκτορας τιμωρείται με -10 για κάθε σύγκρουση αλλά και -1 για κάθε βήμα που περνάει, τότε μπορεί να υιοθετήσει την πολιτική του να συγκρούεται γρήγορα 3 φορές σε κάθε επεισόδιο, ώστε να τερματίζει το επεισόδιο με συνολική ανταμοιβή  $\sim -30$ , αντί για  $\sim -600$ , εφόσον ολοκληρωθεί το επεισόδιο χωρίς να παρκάρει. Ωστόσο, όπως και πριν, αυτή η τεχνική δεν προσέφερε στην πράξη, κάποια βελτίωση στην εκπαίδευση.

#### 5.6.4 Παράμετροι Εκπαίδευσης

Στη βιβλιογραφία, αναφέρεται συχνά η επιρροή των παραμέτρων της εκπαίδευσης στην τελική επιτυχία αυτής. Ωστόσο στην πράξη, δεν παρατηρήσαμε μεγαλές αλλαγές στην απόδοση των πρακτόρων, από τη μεταβολή υπερ-παραμέτρων του αλγορίθμου, όπως ο ρυθμός μάθησης ή ο συντελεστής εντροπίας. Παρόμοια ήταν τα αποτέλεσματα και από την αλλαγή της δομής του νευρωνικού δικτύου. Συγκεκριμένα, τα μικρά νευρωνικά δίκτυα (1-2 κρυφά επίπεδα των 32-64 νευρώνων) θεωρούνται επαρκή για απλές εργασίες, ενώ τα πιο μεγάλα και πολύπλοκα δίκτυα (3-4 κρυφά επίπεδα των 256-512 νευρώνων) κρίνονται απαραίτητα για πιο δύσκολες εργασίες, εισάγοντας όμως τον κίνδυνο της υπερπροσαρμογής. Έτσι, προτείνεται αρχικά η χρήση μικρών δικτύων και η αύξηση της πολυπλοκότητας τους, μόνο αν αποδειχθεί αναγκαίο. Όμως, παρά τις αλλαγές στο μέγεθος των δικτύων που δοκιμάσαμε, δεν πετυχάμε βελτίωση της απόδοσης των πρακτόρων, με αυτόν τον τρόπο.

Βέβαια, οι παρατηρήσεις μας αυτές, ίσως οφείλονται στη χρήση της βιβλιοθήκης Stable-Baselines3, η οποία παρέχει προεπιλεγμένες τιμές για τις υπερ-παραμέτρους των αλγορίθμων της και την αρχιτεκτονική των νευρωνικών δικτύων. Επομένως, αυτές οι παραμέτροι είναι επιλεγμένες με βάση τις τιμές τους στις αρχικές δημοσιεύσεις των αλγορίθμων, καθώς και την εμπειρία των δημιουργών της βιβλιοθήκης. Έτσι, θεωρούνται σε μεγάλο βαθμό βελτιστοποιημένες και προτείνονται από τη βιβλιοθήκη, για τη χρήση των αλγορίθμων της (Raffin 2021). Για αυτό, θα προτείναμε σε μελλοντικές εργασίες που χρησιμοποιούν αξιόπιστες υλοποιήσεις των αλγορίθμων, να επικεντρωθούν στη σωστή μοντελοποίηση του προβλήματος, όπως αυτή εξετάζεται στη συνέχεια, και όχι στις τροποποιήσεις των υπερ-παραμέτρων.

#### 5.6.5 Κανονικοποιήσεις τιμών

Η τεχνική της κανονικοποίησης των τιμών, διαφόρων παραμέτρων της εκπαίδευσης, θεωρείται πως μπορεί να συμβάλει στην αύξηση της απόδοσης των πρακτόρων.

##### Κανονικοποίηση καταστάσεων

Το διάνυσμα κατάστασης αποτελεί την είσοδο του νευρωνικού δικτύου του πράκτορα. Γενικά, τα νευρωνικά δίκτυα λειτουργούν καλύτερα, όταν όλες οι είσοδοι κάθε επίπεδου είναι μικρότερες ή ίσες της μονάδας. Για τα ενδιάμεσα επίπεδα του δικτύου, αυτό εξασφαλίζεται από τις συναρτήσεις ενεργοποίησης. Ωστόσο, για το πρώτο επίπεδο, πρέπει να φροντίσουν οι σχεδιαστές για την κανονικοποίηση της εισόδου του δικτύου, διαιρώντας την κάθε τιμή εισόδου, με τη μέγιστη δυνατή τιμή που μπορεί να λάβει. Για αυτό, στην παρούσα εργασία, κανονικοποιήσαμε τις εισόδους στο διάστημα  $[0, 1]$ , για τους αλγορίθμους που χρησιμοποιούν την συνάρτηση ενεργοποίησης ReLU και

## 5.6 Καλές πρακτικές

---

στο διάστημα  $[-1, 1]$ , για τους αλγορίθμους που χρησιμοποιούν την συνάρτηση ενεργοποίησης Tanh, ώστε τα δεδομένα εισόδου της Tanh να έχουν μέση τιμή 0.

### Κανονικοποίηση ενεργειών

Όταν χρησιμοποιείται συνεχής χώρος ενεργειών (στην περίπτωση μας, στους αλγορίθμους SAC, DDPG, TD3), μία καλή πρακτική είναι η κανονικοποίηση του, ώστε να είναι συμμετρικός σε κάθε ενέργεια. Συνήθως, επιλέγεται η κανονικοποίηση των ενεργειών στο διάστημα  $[-1, 1]$ , όπως έγινε και σε αυτήν την εργασία, καθώς οι περισσότεροι αλγόριθμοι ενισχυτικής μάθησης βασίζονται σε Γκαουσιανή κατανομή με μέση τιμή  $\mu = 0$  και τυπική απόκλιση  $\sigma = 1$ . Επομένως, η έλλειψη κανονικοποίησης του χώρου ενεργειών, μπορεί να βλάψει την εκπαίδευση και είναι δύσκολο να αποσφαλματωθεί (Leger και Raffin 2024).

### Κανονικοποίηση ανταμοιβών

Τέλος, συχνά στη βιβλιογραφία αναφέρεται και η τεχνική της κανονικοποίησης των ανταμοιβών σε κάποιο διάστημα, όπως π.χ. στο  $[0, 1]$ . Ωστόσο, αυτό δεν κρίνεται απαραίτητο, καθώς αυτό που θεωρείται σημαντικό, είναι η σχετική διαφορά μεταξύ των τιμών των ανταμοιβών. Συγκεκριμένα, ανταμοιβές που είναι πολύ μεγαλύτερες σε μέγεθος από τις υπόλοιπες, μπορεί να κυριαρχήσουν στην εκπαίδευση και ο πράκτορας να αφοσιωθεί σε αυτές.

Ωστόσο, στο πρόβλημα της αυτόματης στάθμευσης, όταν η επιβράβευση για τη στάθμευση ήταν συγκρίσιμη με τις υπόλοιπες ανταμοιβές, τότε ο πράκτορας δεν επικεντρωνόταν σε αυτήν, αλλά ανέπτυσσε υποβέλτιστες πολιτικές. Επομένως, κρίναμε σκόπιμο, να αυξήσουμε σε μεγάλο βαθμό την ανταμοιβή της στάθμευσης, ώστε να γίνει σαφές στον πράκτορα, πως αυτή αποτελεί τον βασικό στόχο του. Στη συνέχεια, μετά την επίτευξη αυτού του στόχου από τον πράκτορα, ενθαρρύναμε τη βελτίωση της πολιτικής του, για παράδειγμα αυξάνοντας την τιμωρία των συγκρούσεων. Άρα, η αρχική απόκλιση της ανταμοιβής της στάθμευσης σε σχέση με τις υπόλοιπες, κρίθηκε αναγκαία για την επιτυχία των εκπαιδεύσεων κι έτσι, δεν δοκιμάστηκε η κανονικοποίηση των ανταμοιβών.

### 5.6.6 Παράκαμψη βημάτων

Μία τεχνική η οποία συνηθίζεται στην εκπαίδευση πρακτόρων σε περιβάλλοντα παιχνιδιών, είναι το *FrameSkip*, δηλαδή η παράκαμψη βημάτων από τον πράκτορα. Συγκεκριμένα, η ενέργεια του πράκτορα επαναλαμβάνεται για έναν συγκεκριμένο αριθμό βημάτων, π.χ. για 4 βήματα. Οι ενημερώσεις του δικτύου του πράκτορα συμβαίνουν κανονικά σε κάθε βήμα, όμως μόνο ανά 4 βήματα, ανανεώνεται η ενέργεια του πράκτορα στο περιβάλλον. Με τον τρόπο αυτό, αποφεύγεται η υπερβολικά συχνή, εναλλαγή ενεργειών του πράκτορα (δηλαδή κάθε 1/20 του δευτερολέπτου στην

περίπτωση μας) και επιταχύνεται η διαδικασία της εκπαίδευσης. Στην πράξη, διαπιστώσαμε πως η εφαρμογή της τεχνικής αυτής, εξομάλυνε την οδήγηση του πράκτορα και ήταν καθοριστική, για την τελική επιτυχία της εκπαίδευσης.

Η βιβλιοθήκη Stable-Baselines3 έχει υλοποιημένη μία κλάση για την τεχνική αυτή, η οποία ονομάζεται MaxAndSkipEnv. Ωστόσο, η κλάση αυτή, πέραν από τη λειτουργία του FrameSkip, εκτελεί επίσης και αυτή του Max-Pooling Over Frames, δηλαδή επιστρέφει τη μέγιστη τιμή του κάθε pixel, στα βήματα που παρακάμψαμε. Αυτό είναι κάτι επιθυμητό στην περίπτωση ενός περιβάλλοντος εκπαίδευσης Atari, αλλά όχι στο παιχνίδι της αυτόματης στάθμευσης, καθώς ο χώρος καταστάσεων μας δεν αποτελείται από την εικόνα του παιχνιδιού (*pixel data*). Για αυτό τροποποιήσαμε κατάλληλα τον κώδικα της κλάσης MaxAndSkipEnv, ώστε να εφαρμόζεται μόνο η τεχνική του FrameSkip.

### 5.6.7 Επίπεδα δυσκολίας και Κλιμακωτή Μάθηση

Συχνά, είναι δύσκολο να επιτύχει ο πράκτορας απευθείας την επιθυμητή συμπεριφορά, σε περιβάλλοντα με υψηλό βαθμό δυσκολίας. Για αυτό, προτείνεται η εκπαίδευση να ξεκινάει από απλοποιημένες εκδοχές του περιβάλλοντος (*toy problems*), με σκοπό να δείξει σε αυτές ο πράκτορας ορισμένα σημάδια ζωής, δηλαδή κάποια πρώτα καλά αποτελέσματα, που αποδεικνύουν πως ο αλγόριθμος λειτουργεί σωστά (Schulman 2017). Στη συνέχεια, ο σχεδιαστής μπορεί να αυξήσει σταδιακά το βαθμό δυσκολίας του περιβάλλοντος, μέχρι να φτάσει στο επιθυμητό επίπεδο.

Πράγματι, αυτή η προσέγγιση αποδείχθηκε πολύ χρήσιμη σε αυτήν την εργασία. Η δημιουργία των 4 επιπέδων δυσκολίας του παιχνίδιου (βλ. 4.2.3), συνέβαλε σημαντικά στην ευκολότερη αποσφαλμάτωση της κάθε εκπαίδευσης. Έπειτα, αφού ο πράκτορας είχε φτάσει σε ικανοποιητικό επίπεδο απόδοσης σε εύκολα επίπεδα, εφαρμόστηκε η τεχνική της Κλιμακωτής Μάθησης (*Curriculum Learning*) και ήταν καθοριστική για την επίτευξη της επιθυμητής συμπεριφοράς στα δυσκολότερα επίπεδα.

Συγκεκριμένα, η Κλιμακωτή Μάθηση είναι μία ειδική κατηγορία της Μεταφοράς Γνώσης (*Transfer Learning*), η οποία περιγράφει το πώς μπορεί η εμπειρία ενός πράκτορα σε μία εργασία μάθησης να τον βοηθήσει να μάθει καλύτερα, κάποια άλλη, σχετική εργασία (Russell και Norvig 2021). Για παράδειγμα, ένας πράκτορας ρομπότ που έχει μάθει να ποιέσει τέννις, θα είναι σε θέση να εκπαιδευτεί πιο εύκολα να παίζει ένα παρόμοιο παιχνίδι, όπως το πινγκ-πονγκ. Η Κλιμακωτή Μάθηση στηρίζεται σε αυτή την ιδέα και προτείνει την διαδοχική εκπαίδευση του πράκτορα, σε περιβάλλοντα αυξανόμενης δυσκολίας. Για παράδειγμα, στην περίπτωση μας, κανένας πράκτορας δεν κατάφερε να εκπαιδευτεί επιτυχώς, απευθείας στο δυσκολότερο επίπεδο του παιχνιδιού (επίπεδο 4), το οποίο εισάγει τη συνθήκη του χρονικού διαστήματος ακινησίας για τον καθορισμό της επιτυχούς στάθμευσης. Ωστόσο, οι πράκτορες που είχαν εκπαιδευτεί επιτυχώς στο επίπεδο 3 και είχαν μάθει να εισέρχονται εντός της θέσης στάθμευσης, με την επανεκπαίδευση τους στο επίπεδο 4, κατάφεραν να προσαρμόσουν τη γνώση τους, ώστε να μένουν πλέον ακίνητοι εντός της θέσης.

### 5.6.8 Παρακολούθηση μετρικών

Η παρακολούθηση των γραφικών παραστάσεων, που σχεδιάζονται αυτόματα από το εργαλείο Tensorboard, είναι κρίσιμη για τη διαδικασία της εκπαίδευσης. Μέσω αυτών, λαμβάνεται η απόφαση για συνέχιση ή μη μίας εκπαίδευσης. Αυτό καταδεικνύεται συνήθως, από την τάση της γραφικής των ανταμοιβών, καθώς αυτή φανερώνει τις αλλαγές στην πολιτική του πράκτορα. Όταν η γραφική αυτή, έχει σχετικά σταθερή τιμή για ένα μεγάλο χρονικό διάστημα (π.χ. επί 1M steps), θεωρείται πως η εκπαίδευση έχει συγκλίνει και μπορεί να σταματήσει. Τότε, είναι ασφαλής η αξιολόγηση του πράκτορα, καθώς πρόκειται για την τελική απόδοση του (Patterson κ.ά. 2023). Μάλιστα, από την εμπειρία μας, παρατηρήσαμε πως οι περισσότεροι αλγόριθμοι συγκλίνουν στα πρώτα 2-4M steps εκπαίδευσης, ενώ μετά από αυτό το σημείο, η αύξηση της απόδοσης είναι ελάχιστη.

Επίσης, οι μετρικές του Tensorboard μπορεί να υποδείξουν τις τροποποίησεις που πρέπει να γίνουν στην εκπαίδευση. Για παράδειγμα, όταν η καμπύλη των ανταμοιβών εμφανίζει μεγάλες διακυμάνσεις, τότε αυτό αποτελεί ένδειξη πως ο ρυθμός μάθησης είναι πολύ υψηλός και πρέπει να μειωθεί, προκειμένου να επιτευχθεί πιο σταθερή εκπαίδευση. Αντίστοιχα, στην περίπτωση όπου η καμπύλη των ανταμοιβών συγκλίνει πολύ νωρίς, τότε ίσως χρειάζεται η ρύθμιση της εντροπίας του αλγορίθμου (*entropy regularization*). Πιο αναλυτικά, η αύξηση του συντελεστή εντροπίας του αλγορίθμου ενθαρρύνει την εξερεύνηση του πράκτορα, καθώς ωθεί σε πιο ίση κατανομή των ενεργειών του. Ωστόσο, ένας υπερβολικά μεγάλος συντελεστής εντροπίας, θα προκαλέσει απλά την τυχαιότητα των ενεργειών του πράκτορα.

Τέλος, είναι προτιμότερο οι μετρικές να σχεδιάζονται με βάση τα βήματα εκπαίδευσης και όχι τα επεισόδια εκπαίδευσης. Με αυτόν τον τρόπο, εξασφαλίζεται πως διαφορετικές εκπαιδεύσεις συγκρίνονται στον ίδιο αριθμό δειγμάτων των πρακτόρων, ανεξαρτήτως των μηκών των επεισοδίων τους.

### 5.6.9 Διατήρηση αρχείου εκπαιδεύσεων

Μιά σημαντική διάσταση της διαδικασίας των εκπαιδεύσεων, είναι η συνεπής διατήρηση ενός αναλυτικού αρχείου. Η μέθοδος αυτή γίνεται απαραίτητη, όταν το πλήθος των εκπαιδεύσεων αυξάνεται και η διάρκεια τους μεγαλώνει (Amid 2018). Τότε, ένα λεπτομερές αρχείο θα βοηθήσει τον σχεδιαστή να οργανώσει καλύτερα τις σκέψεις του και να μην ξεχνάει ποιές ιδέες έχουν δοκιμαστεί ήδη και ποιά ήταν τα αποτελέσματα τους.

Στο πρόβλημα της αυτόματης στάθμευσης, κρατήσαμε διαφορετικό αρχείο για κάθε αλγόριθμο που εξετάστηκε. Τα αρχεία αυτά βρίσκονται στο [αποθετήριο](#) της εργασίας, στον φάκελο `parking_game/Saved-training/Όνομα-αλγορίθμου`. Σε κάθε φάκελο, βρίσκονται τα εξής στοιχεία:

- Υποφάκελοι για κάθε ξεχωριστή εκπαίδευση, οι οποίοι περιέχουν το αντίστοιχο αρχείο κώδικα,

τις γραφικές παραστάσεις του Tensorboard και τα αρχεία .zip με τα βάρη των καλύτερων πρακτόρων

- Το αρχείο `Όνομα-Αλγορίθμου.txt`, το οποίο περιέχει τις εξής πληροφορίες για κάθε εκπαίδευση:
  - τι καινούργιο δοκιμάστηκε στη συγκεκριμένη εκπαίδευση και που αποσκοπεί
  - τα βήματα εκπαίδευσης
  - μία σύντομη, λεκτική περιγραφή της εικόνας των μετρικών
  - συμπεράσματα από την εξέταση του πράκτορα στο παιχνίδι

Η διαδικασία αυτή, αν και χρονοβόρα, αποδείχθηκε τελικά κρίσιμη, για την επιτυχία της εργασίας και για αυτό την προτείνουμε ανεπιφύλακτα σε μελλοντικούς ερευνητές.

## 5.7 Μοντελοποίηση προβλήματος

Σε αυτήν την ενότητα παρουσιάζεται η τελική μοντελοποίηση του προβλήματος αυτόματης στάθμευσης, μέσω της οποίας επιτεύχθηκαν τα καλύτερα αποτελέσματα των πρακτόρων. Συγκεκριμένα, θα εξεταστεί πρώτα, η αρχιτεκτονική των νευρωνικών δικτύων και στη συνέχεια, θα αναλυθεί η συνάρτηση ανταμοιβής.

### 5.7.1 Αρχιτεκτονική νευρωνικών δικτύων

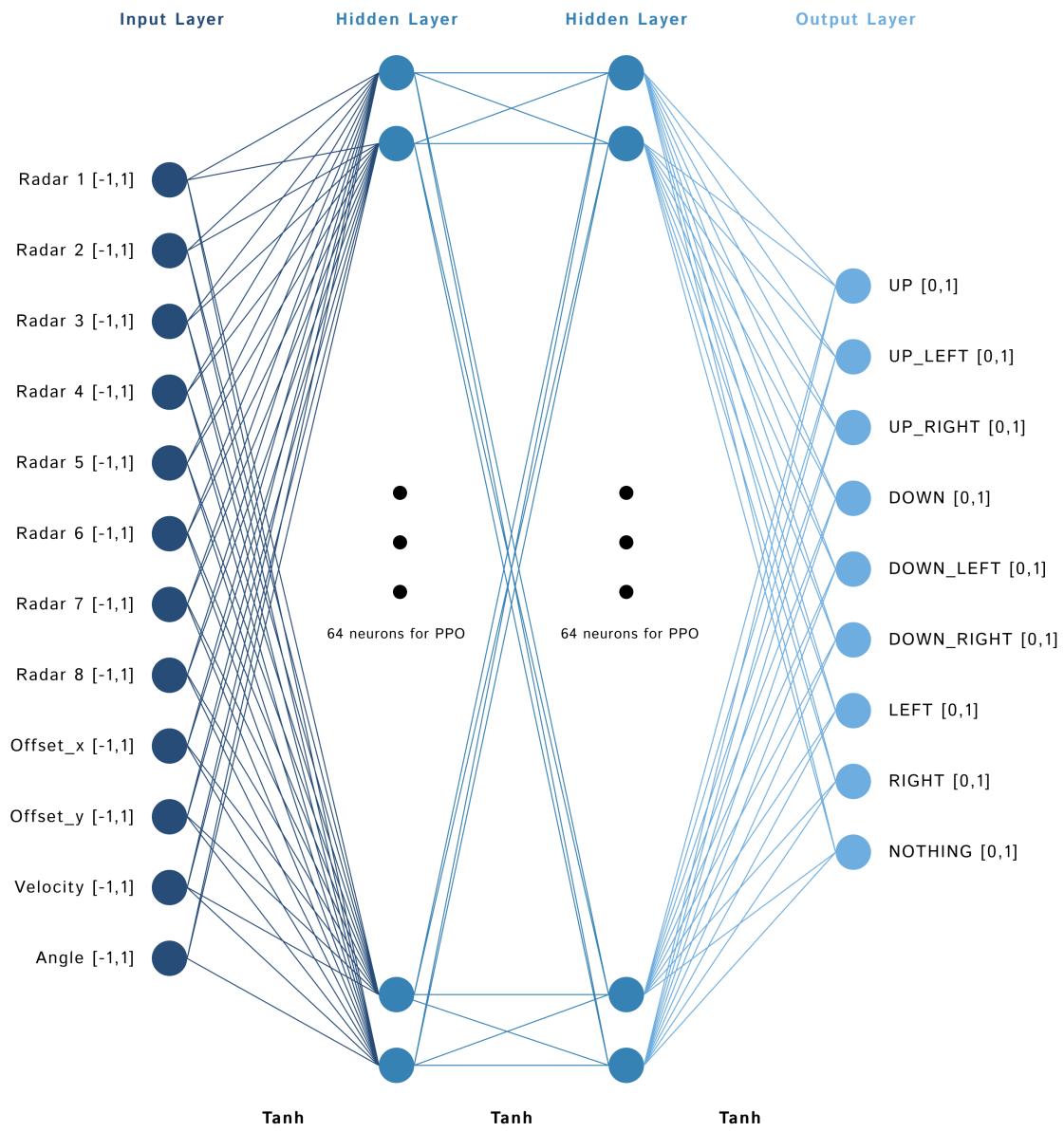
#### Διακριτός χώρος ενεργειών

Η τελική αρχιτεκτονική του νευρωνικού δικτύου, για την περίπτωση των αλγορίθμων με διακριτό χώρο ενεργειών (PPO), φαίνεται στην *Εικόνα 5.9*:

Αρχικά, παρατηρούμε πως το νευρωνικό δίκτυο αποτελείται από 2 κρυφά επίπεδα, των 64 νευρώνων το καθένα, όπως και στην αρχική δημοσίευση του αλγορίθμου. Στη συνέχεια, αξίζει να σταθούμε στην κατάρτιση του χώρου καταστάσεων, δηλαδή της εισόδου του πράκτορα, που απαρτίζεται από τα εξής 12 στοιχεία:

- **Radar 1-8:** οι ακτίνες ραντάρ λειτουργούν ως αισθητήρες για το περιβάλλον του πράκτορα, παρέχοντας τις αποστάσεις του από άλλα αντικείμενα σε 8 κατευθύνσεις.
- **Offset\_x, Offset\_y:** αποτελούν την σχετική απόσταση του πράκτορα, από την ελεύθερη θέση στάθμευσης. Επιλέχθηκαν αντί της απλής απόστασης πράκτορα-θέσης, καθώς έτσι, πέραν από την πληροφορία του πόσο απέχει ο πράκτορας από τον στόχο του, παρέχεται σε αυτόν και η κατεύθυνση προς την οποία πρέπει να κινηθεί. Για παράδειγμα, όταν η είσοδος Offset\_x είναι αρνητική, αυτό σημαίνει πως ο πράκτορας βρίσκεται στα αριστερά της θέσης στάθμευσης, ενώ όταν η είσοδος αυτή μηδενιστεί, ο πράκτορας βρίσκεται στον ίδιο οριζόντιο άξονα με

## Neural Network Architecture for Discrete Action Space



**Εικόνα 5.9.** Αρχιτεκτονική νευρωνικού δικτύου αλγορίθμων με διακριτό χώρο ενεργειών.

τη θέση στάθμευσης. Επομένως, οι είσοδοι Offset\_x και Offset\_y βοηθούν τον πράκτορα να αναγνωρίσει την κατεύθυνση προς την οποία πρέπει να κινηθεί (π.χ. δεξιά και πάνω), καθώς και να κατανοήσει πως ο στόχος του θα επιτευχθεί, όταν οι δύο αυτές τιμές εισόδων συγκλίνουν στο 0.

- **Velocity:** η ταχύτητα του πράκτορα τον ενημερώνει αφενός για το πόσο γρήγορα κινείται και αφετέρου, για το αν κινείται προς τα εμπρός ή προς τα πίσω. Έτσι, η συγκεκριμένη είσοδος βοηθάει τον πράκτορα να προβλέψει πότε πρόκειται να συγκρουστεί με κάποιο αντικείμενο, καθώς και να κατανοήσει πως ο στόχος της στάθμευσης θα επιτευχθεί, όταν η είσοδος αυτή μηδενιστεί.
- **Angle:** η γωνία του πράκτορα, ως προς τον άξονα x, τον ενημερώνει για τον προσανατολισμό του. Επομένως, όπως και πριν, η είσοδος αυτή βοηθάει τον πράκτορα να προβλέψει πότε πρόκειται να συγκρουστεί με κάποιο αντικείμενο, καθώς και να κατανοήσει πως ο στόχος του θα επιτευχθεί, όταν η συγκεκριμένη είσοδος συγκλίνει είτε στην τιμή 0 (το αυτοκίνητο είναι στραμμένο προς τα κάτω), είτε στην τιμή  $\pm 1$  (το αυτοκίνητο είναι στραμμένο προς τα πάνω).

Κάθε μία από τις παραπάνω εισόδους έχει υποστεί κανονικοποίηση, ώστε να βρίσκεται στο διάστημα [-1, 1], καθώς, όπως βλέπουμε και στην Εικόνα 5.9, το επίπεδο εισόδου του δικτύου χρησιμοποιεί την συνάρτηση ενεργοποίησης Tanh.

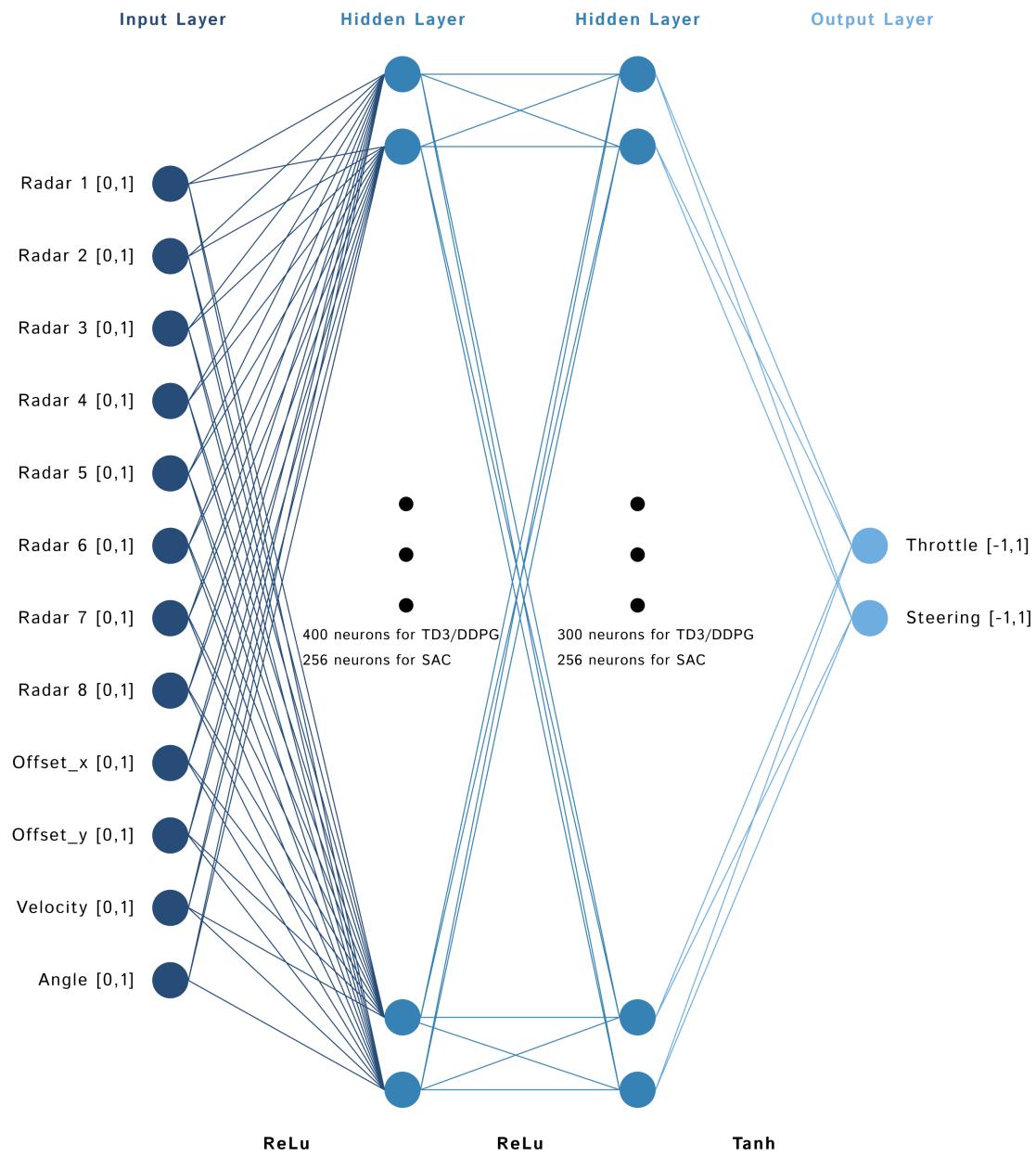
Τέλος, ο χώρος ενεργειών αποτελείται από 9 διακριτές ενέργειες. Η δράση της κάθε ενέργειας είναι αυτονόητη από το όνομα της και αντιστοιχεί στην πίεση κανενός, ενός ή δύο βελών του πληκτρολογίου. Αξίζει μόνο να σημειώσουμε, πως οι ενέργειες LEFT και RIGHT προκαλούν απλώς την περιστροφή του αυτοκινήτου προς την αντίστοιχη κατεύθυνση και είναι χρήσιμες, μόνο όταν το αυτοκίνητο έχει ήδη, κάποια αρχική ταχύτητα. Αντίθετα, όταν το αυτοκίνητο είναι ακίνητο, οι ενέργειες αυτές δεν θα έχουν κανένα αποτέλεσμα. Ο αλγόριθμος PPO επιλέγει μία τιμή στο διάστημα [0, 1] για κάθε ενέργεια, η οποία υποδηλώνει την πιθανότητα επιλογής της. Κατά την εκπαίδευση, χρησιμοποιείται στοχαστική πολιτική και οι ενέργειες επιλέγονται με βάση αυτήν την κατανομή πιθανοτήτων, ενώ κατά την αξιολόγηση, χρησιμοποιείται ντετερμινιστική πολιτική και επιλέγεται απλώς η ενέργεια με τη μεγαλύτερη πιθανότητα.

### Συνεχής χώρος ενεργειών

Η τελική αρχιτεκτονική του νευρωνικού δικτύου, για την περίπτωση των αλγορίθμων με συνεχή χώρο ενεργείων (SAC, TD3, DDPG), φαίνεται στην Εικόνα 5.10:

Αρχικά, οι αλγόριθμοι που χρησιμοποιούνται σε αυτήν την περίπτωση ανήκουν στην κατηγορία δράστη-κριτή, επομένως το νευρωνικό δίκτυο που εξετάζουμε είναι αυτό του δράστη. Όπως και πριν, το δίκτυο αποτελείται από 2 κρυφά επίπεδα, όμως τώρα το πλήθος των νευρώνων σε αυτά, είναι πολύ μεγαλύτερο. Συγκεκριμένα, ο αλγόριθμος SAC χρησιμοποιεί 256 νευρώνες σε κάθε κρυφό επίπεδο,

## Neural Network Architecture for Continuous Action Space (Actor)



**Εικόνα 5.10.** Αρχιτεκτονική νευρωνικού δικτύου αλγορίθμων με συνεχή χώρο ενεργειών.

ενώ οι αλγόριθμοι TD3 και DDPG χρησιμοποιούν 400 νευρώνες στο πρώτο κρυφό επίπεδο και 300 στο δεύτερο. Οι τιμές αυτές είναι οι προκαθορισμένες από τη βιβλιοθήκη Stable-Baselines3.

Οι είσοδοι του δικτύου είναι ίδιες με πριν, με τη διαφορά πως πλέον, κανονικοποιούνται στο διάστημα [0, 1]. Αντίθετα, οι έξοδοι έχουν μειωθεί σε μόλις δύο: μία που αντιστοιχεί στην ισχύ του πράκτορα (THROTTLE) και μία που αντιστοιχεί στον προσανατολισμό του (STEERING). Οι δύο αυτές έξοδοι, λαμβάνουν τιμές στο διάστημα [-1, 1]. Συγκεκριμένα, κάθε αλγόριθμος εξάγει δύο τιμές, για κάθε ενέργεια: τη μέση τιμή της ενέργειας ( $\mu$ ) και την τυπική απόκλιση της ( $\sigma$ ). Όταν η πολιτική του αλγορίθμου είναι στοχαστική (SAC κατά την εκπαίδευση), η επιλογή της τελικής τιμής κάθε ενέργειας γίνεται μέσω δειγματοληψίας, από την κατανομή που περιγράφεται από τη μέση τιμή και την τυπική απόκλιση. Αντίθετα, όταν η πολιτική είναι ντετερμινιστική (SAC κατά την αξιολόγηση, TD3, DDPG), χρησιμοποιείται απλά η μέση τιμή, ως τελική τιμή της ενέργειας. Βέβαια, αφού προκύψουν οι τελικές τιμές για τις δύο αυτές εξόδους, ο κώδικας μας θα τις μεταφράσει σε μία από τις 9 διακριτές ενέργειες, που αναφέρθηκαν παραπάνω. Μέσω της μετατροπής αυτής, οι αλγόριθμοι που χρησιμοποιούν συνεχή χώρο ενεργείων, ελέγχουν κι αυτοί το αυτοκίνητο με τα βέλη του πληκτρολογίου, όπως θα έκανε ένας άνθρωπος που παίζει το παιχνίδι.

## 5.7.2 Συνάρτηση ανταμοιβής

### Αραιές ανταμοιβές - αλγόριθμος PPO

Η συνάρτηση ανταμοιβής που πέτυχε την καλύτερη επίδοση των πρακτόρων του αλγορίθμου PPO, στο επίπεδο δυσκολίας 3 - άμεση στάθμευση, δίνεται παρακάτω, υπό μορφή ψευδοκώδικα:

---

#### Reward Function 1 - Instant Parking: PPO

---

```

1: reward ← 0
2: if car is inside spot then
3:   reward ← reward + 500
4: else
5:   reward ← reward - 3           ▷ Punish the car for being away from parking spot
6:   if car collides then
7:     reward ← reward - 10
8:   end if
9:   if car is moving then
10:    reward ← reward + 2
11:  end if
12: end if

```

---

**Εικόνα 5.11.** Συνάρτηση με αραιές ανταμοιβές - Άμεση στάθμευση.

Παρατηρούμε αρχικά, πως οι ανταμοιβές σε αυτήν την περίπτωση είναι αραιές, καθώς η συνάρτηση

ανταμοιβής δεν οδηγεί τον πράκτορα προς τον στόχο, αλλά απλώς τον επιβραβέυει, όταν φτάσει σε αυτόν (+500). Βέβαια, υπάρχουν κάποιες δευτερεύουσες ανταμοιβές, όπως η επιβράβευση του πράκτορα όταν κινείται (+2), προκειμένου να τον ενθαρρύνουμε να εξερευνήσει το περιβάλλον, αλλά και η τιμωρία του, όταν συγκρούεται με κάποιο αντικείμενο (-10), ώστε να τον αθήσουμε να βελτιώσει την πολιτική του. Ακόμα, η τιμωρία σε κάθε βήμα (-3) είναι απαραίτητη, καθώς χωρίς αυτήν, ο πράκτορας πετύχαινε θετική ανταμοιβή μόνο με την κίνηση του, κι έτσι έμαθε να κάνει πρώτα έναν γύρο του χάρτη και μετά να παρκάρει (reward hacking).

Ωστόσο, όταν δοκιμάσαμε να επανεκπαίδευσουμε αυτόν τον πράκτορα στο επόμενο επίπεδο δυσκολίας, αυτό της κανονικής στάθμευσης (με χρονικό όριο 2 δευτερολέπτων), τροποποιώντας την παραπάνω συνάρτηση ανταμοιβής, τα αποτελέσματα δεν ήταν ικανοποιητικά και ο πράκτορας δεν κατάφερε να μάθει να σταθμεύει αξιόπιστα.

### Διαμόρφωση ανταμοιβής - αλγόριθμοι SAC, TD3, DDPG

Η συνάρτηση ανταμοιβής που πέτυχε την καλύτερη επίδοση των πρακτόρων των αλγορίθμων SAC, TD3 και DDPG, στο επίπεδο δυσκολίας 3 - άμεση στάθμευση, δίνεται στην επόμενη σελίδα, υπό μορφή ψευδοκώδικα:

Παρατηρούμε αρχικά, πως οι ανταμοιβές σε αυτήν την περίπτωση είναι πολύ πιο συχνές, καθώς η συνάρτηση ανταμοιβής κατευθύνει τον πράκτορα προς τον στόχο. Έτσι, ο πράκτορας τιμωρείται (-6) όσο πιο μακριά βρίσκεται από την ελεύθερη θέση στάθμευσης, στον όξονα x και στον όξονα y. Επιπλέον, ο πράκτορας τιμωρείται όταν μένει ακίνητος (-5), προκειμένου να τον ενθαρρύνουμε να εξερευνήσει το περιβάλλον, όπως και πριν. Όμως, σε αντίθεση με πριν, πλέον, ο πράκτορας τιμωρείται ακόμα κι όταν κινείται με μικρή ταχύτητα. Αυτή η τιμωρία προέκυψε, καθώς χωρίς αυτήν, ο πράκτορας έμαθε να αποφεύγει την τιμωρία της ακινησίας, κινούμενος πολύ αργά (υποβέλτιστη πολιτική). Μάλιστα, η τιμωρία για την κίνηση με μικρή ταχύτητα είναι μεγαλύτερη (-5), όταν ο πράκτορας βρίσκεται μακριά από τον στόχο του και μικρότερη (-3), όταν βρίσκεται κοντά σε αυτόν. Η επιλογή αυτή, έγινε με το σκεπτικό πως είναι αναμενόμενο, ο πράκτορας να επιβραδύνει όταν βρίσκεται κοντά στη θέση στάθμευσης, προκειμένου να πραγματοποιήσει τους απαραίτητους ελιγμούς για να παρκάρει. Επομένως, στην περίπτωση αυτή, ο πράκτορας τιμωρείται λιγότερο. Επίσης, όταν ο πράκτορας βρίσκεται κοντά στη θέση στάθμευσης, επιβραβεύεται όσο έχει την κατάλληλη γωνία (+0.5), προκειμένου να τον καθοδηγήσουμε να παρκάρει. Ακόμα, όπως και πριν, ο πράκτορας τιμωρείται (-10) όταν συγκρούεται με κάποιο αντικείμενο, ώστε να μάθει να αποφεύγει τις συγκρούσεις. Τέλος, ο πράκτορας λαμβάνει μεγάλη επιβράβευση, όταν εισέρχεται στην ελεύθερη θέση στάθμευσης (+5000).

---

Reward Function 2 - Instant Parking: SAC, TD3, DDPG

---

```

1: reward ← 0
2: if car is inside parking spot then
3:   reward ← reward + 5000
4:   if car entered parking spot moving forward then
5:     reward ← reward + 1000                                ▷ Extra reward
6:   end if
7: else
8:   reward ← reward − (offset_x × 6)                  ▷ Punishment based on
9:   reward ← reward − (offset_y × 6)                  ▷ distance from x,y axis
10:  if car is moving forward then
11:    reward ← reward + 1
12:  end if
13:  if car collides then
14:    reward ← reward − 10
15:  end if
16:  if car is away from parking spot then
17:    if car is moving too slow then
18:      reward ← reward − 5
19:      if car is not moving then
20:        reward ← reward − 5
21:      end if
22:    end if
23:  else                                                 ▷ When the car is near the parking spot
24:    if car is moving too slow then
25:      reward ← reward − 3
26:      if car is not moving then
27:        reward ← reward − 5
28:      end if
29:    end if
30:    if car is in the right angle then
31:      reward ← reward + 0.5
32:    end if
33:  end if
34: end if

```

---

**Εικόνα 5.12.** Συνάρτηση με διαμόρφωση ανταμοιβής - Άμεση στάθμευση.

## 5.7 Μοντελοποίηση προβλήματος

---

Οι τιμές των ανταμοιβών είναι ελαφρώς διαφορετικές στους τρεις αλγορίθμους, καθώς έχουν προσαρμοστεί στη συμπεριφορά του κάθε πράκτορα. Επίσης, υπάρχουν δύο κομμάτια της συνάρτησης ανταμοιβής που διαφέρουν στους τρεις αλγορίθμους:

- το πρώτο κομμάτι είναι η επιπλέον επιβράβευση (+1000) για την είσοδο του πράκτορα στη θέση στάθμευσης, κινούμενος προς τα εμπρός (γραμμές 4-5). Αυτό το «bonus», προστέθηκε στους αλγορίθμους που κρίθηκε απαραίτητο, προκειμένου ο πράκτορας να μάθει να παρκάρει τόσο προς τα εμπρός, όσο και με την όπισθεν.
- το δεύτερο κομμάτι, αφορά την επιπλέον επιβράβευση (+1) για την κίνηση του πράκτορα προς τα εμπρός (γραμμές 10-11). Παρόμοια με πριν, αυτή η επιβράβευση προστέθηκε σε συγκεκριμένους αλγορίθμους, καθώς χωρίς αυτήν, οι πράκτορες τους έμαθαν να κινούνται μόνο προς τα πίσω.

Είναι πιθανόν, αυτές οι μικρές αλλαγές στη συμπεριφορά των πρακτόρων, να οφείλονται στην εξάρτηση των εκπαιδεύσεων από τους ψευδο-τυχαίους αριθμούς του περιβάλλοντος (βλ. υποενότητα [5.5.5](#)). Επομένως, οι επιμέρους τροποποίησεις της συνάρτησης ανταμοιβής σε κάθε αλγόριθμο, έχουν μικρή σημασία. Εξάλλου, η γενική δομή της συνάρτησης ανταμοιβής είναι η ίδια, σε όλες τις περιπτώσεις.

Στη συνέχεια, επανεκπαιδεύσαμε τους πράκτορες των αλγορίθμων SAC, TD3 και DDPG, στο επόμενο και τελευταίο επίπεδο δυσκολίας, τροποποιώντας την παραπάνω συνάρτηση ανταμοιβής. Η νέα συνάρτηση ανταμοιβής, για το επίπεδο δυσκολίας 4 - κανονική στάθμευση (με χρονικό όριο 2 δευτερολέπτων), δίνεται στην επόμενη σελίδα, υπό μορφή ψευδοκώδικα:

Πλέον, εφαρμόζουμε κλιμακωτή μάθηση και πρέπει να καθοδηγήσουμε τους προηγούμενους πράκτορες, οι οποίοι έχουν μάθει να κινούνται εντός της ελεύθερης θέσης στάθμευσης, να παραμείνουν ακίνητοι εντός αυτής. Για αυτό, στη συνάρτηση ανταμοιβής έχουν προστεθεί οι γραμμές 5-9. Στις συγκεκριμένες γραμμές, ο πράκτορας επιβραβεύεται όσο βρίσκεται εντός της θέσης στάθμευσης (+10), όμως η επιβράβευση αυτή, είναι αντιστρόφως ανάλογη της ταχύτητας του. Επομένως, όσο πιο αργά κινείται ο πράκτορας, τόσο μεγαλύτερη είναι η επιβράβευση. Μάλιστα, όταν η ταχύτητα του πράκτορα μηδενίστει, τότε προστίθεται κι άλλη επιβράβευση (+10). Με αυτόν τον τρόπο, ο πράκτορας σταδιακά μαθαίνει να ακινητοποιείται εντός της θέσης στάθμευσης και όταν το πετύχει αυτό για 2 συνεχόμενα δευτερόλεπτα, τότε λαμβάνει τη μεγάλη επιβράβευση της επιτυχούς στάθμευσης (+5000).

Πράγματι, όπως θα δούμε στο επόμενο κεφάλαιο, μέσω αυτής της συνάρτησης ανταμοιβής, οι πράκτορες των αλγορίθμων SAC, TD3 και σε μικρότερο βαθμό, του DDPG, κατάφεραν να μάθουν να σταθμεύουν αξιόπιστα, στο τελικό επίπεδο δυσκολίας.

---

Reward Function 2 - Normal Parking: SAC, TD3, DDPG

---

```

1: reward  $\leftarrow$  0
2: if car has parked then
3:   reward  $\leftarrow$  reward + 5000
4: else
5:   if car is inside spot then
6:     reward  $\leftarrow$  reward +  $\frac{10}{\text{car's velocity}}$                                  $\triangleright$  Encourage the agent to slow down
7:     if car is stationary then
8:       reward  $\leftarrow$  reward + 10
9:     end if
10:    else
11:      reward  $\leftarrow$  reward - (offset_x  $\times$  6)                                 $\triangleright$  Punishment based on
12:      reward  $\leftarrow$  reward - (offset_y  $\times$  6)                                 $\triangleright$  distance from x,y axis
13:      if car is moving forward then
14:        reward  $\leftarrow$  reward + 1
15:      end if
16:      if car collides then
17:        reward  $\leftarrow$  reward - 10
18:      end if
19:      if car is away from parking spot then
20:        if car is moving too slow then
21:          reward  $\leftarrow$  reward - 5
22:          if car is not moving then
23:            reward  $\leftarrow$  reward - 5
24:          end if
25:        end if
26:      else                                               $\triangleright$  When the car is near the parking spot
27:        if car is moving too slow then
28:          reward  $\leftarrow$  reward - 3
29:          if car is not moving then
30:            reward  $\leftarrow$  reward - 5
31:          end if
32:        end if
33:        if car is in the right angle then
34:          reward  $\leftarrow$  reward + 0.5
35:        end if
36:      end if
37:    end if
38:  end if

```

---

**Εικόνα 5.13.** Συνάρτηση με διαμόρφωση ανταμοιβής - Κανονική στάθμευση.

## 5.8 Εκπαιδεύσεις με τον αλγόριθμο Q-Learning

### 5.8.1 Επισκόπηση εκπαιδεύσεων

Ο αλγόριθμος Q-Learning ήταν ο πρώτος αλγόριθμος που εξετάστηκε και επιλέχθηκε για την απλότητα του, ως μία εισαγωγή στον χώρο της ενισχυτικής μάθησης. Συνολικά, διεξήχθησαν 55 εκπαιδεύσεις του αλγορίθμου, οι οποίες διήρκησαν αθροιστικά 2 ημέρες, 15 ώρες και 45 λεπτά.

Το πρόβλημα που αντιμετωπίσαμε, είναι το πλήθος καταστάσεων του παιχνιδιού. Συγκεκριμένα, το παιχνίδι που αναπτύξαμε αποτελεί ένα περιβάλλον με συνεχή χώρο καταστάσεων. Όμως, προκειμένου να εφαρμόσουμε τον αλγόριθμο Q-Learning σε αυτό, έπρεπε να διακριτοποιήσουμε τον χώρο του παιχνιδιού. Ωστόσο, με την πρώτη διακριτοποίηση που δοκιμάσαμε, το μέγεθος του πίνακα Q προέκυψε 6.5 TB. Επομένως, προβήκαμε σε διαδοχικές διακριτοποιήσεις, έως ότου καταλήξαμε σε αυτήν που φαίνεται στην Εικόνα 5.14, η οποία δημιουργεί πίνακα Q μεγέθους 36 KB.

	Continuous State Space	Discrete State Space	
Radar 1 :	[0, 200]	[0, 1]	
Radar 2 :	[0, 200]	[0, 1]	
Radar 3 :	[0, 200]	[0, 1]	
Radar 4 :	[0, 200]	[0, 1]	
Offset_x :	[-552, 552]	[-1, 0, 1]	
Offset_y :	[-478, 478]	[-1, 0, 1]	
Velocity :	[-1, 2]	[-1, 0, 1]	
Angle :	[0, 360]	[-3, -2, -1, 0, 1, 2, 3]	
			
$2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 3 \times 7 = 3024 \text{ states}$ $3024 \text{ states} \times 7 \text{ actions} = 21168 \text{ Q-table items}$			

Εικόνα 5.14. Τελική διακριτοποίηση χώρου καταστάσεων.

Παρατηρούμε αρχικά, πως προκειμένου να μειώσουμε τις διαστάσεις του πίνακα Q, περικόψαμε κάποιες από τις εισόδους και εξόδους του πράκτορα. Συγκεκριμένα, πλέον υπάρχουν 4 αισθητήρες αντί για 8 και 7 ενέργειες αντί για 9, καθώς αφαιρέθηκαν οι ενέργειες LEFT και RIGHT. Ακόμα, η διακριτοποίηση συνοψίζει την πληροφορία του περιβάλλοντος. Έτσι, οι αισθητήρες πλέον παίρνουν μόνο 2 τιμές: 0 όταν δεν εντοπίζουν κάποιο αντικείμενο και 1 όταν το εντοπίζουν. Αντίστοιχα, οι σχετικές αποστάσεις του πράκτορα από τη θέση στάθμευσης παίρνουν 3 τιμές, για παράδειγμα για τον άξονα x: -1 όταν ο πράκτορας βρίσκεται αριστερά της θέσης, 0 όταν βρίσκεται στον ίδιο οριζόντιο άξονα με τη θέση (με μία απόκλιση  $\pm 10\text{px}$ ) και 1 όταν βρίσκεται δεξιά της. Παρόμοια, η ταχύτητα παίρνει μόνο 3 δυνατές τιμές και η γωνία 7. Οι τιμές αυτές κρίθηκαν οι απολύτως αναγκαίες, ώστε να έχει ο πράκτορας επαρκή πληροφορία, για να λύσει το πρόβλημα. Παρόλα αυτά, το πλήθος στοιχείων του πίνακα Q (21168), παραμένει αρκετά μεγαλύτερο, από αυτό που συνηθίζεται. Για παράδειγμα, στην εργασία (Patel, Carver, και Rahimi 2011), ο πίνακας Q αποτελείται από μόλις 96 στοιχεία, ενώ στην εργασία των (Blomqvist και Andersson 2022), το μέγεθος του πίνακα Q είναι 576. Επομένως, παρά τις διακριτοποιήσεις που κάναμε, το πρόβλημα παραμένει αρκετά μεγάλο, ενώ

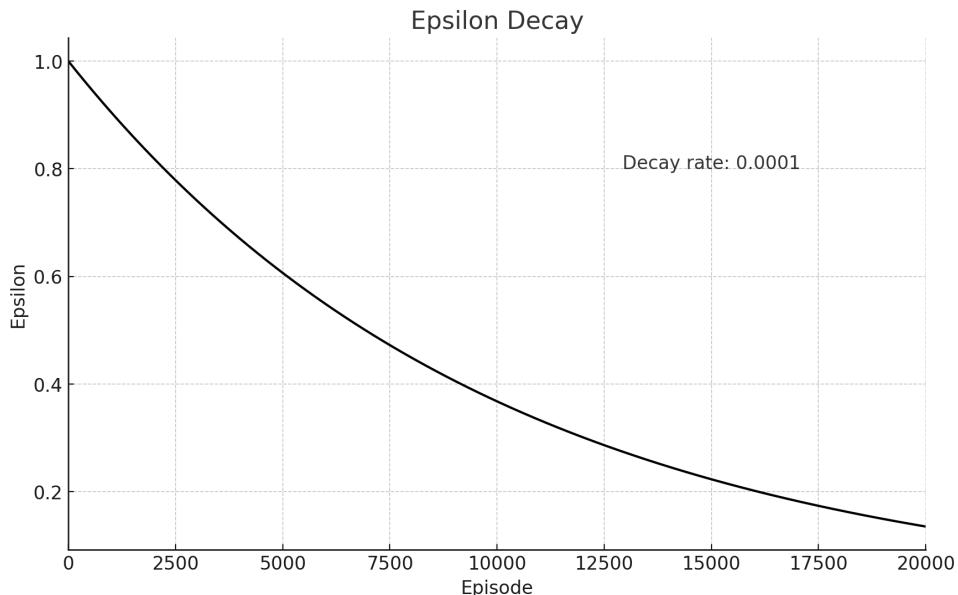
είναι πιθανή η δημιουργία νέων προβλημάτων, λόγω της έλλειψης ακρίβειας.

Αξίζει να σημειωθεί, πως εξετάστηκε και η τεχνική της διακριτοποίησης με λογαρίθμηση, όσον αφορά την απόσταση του πράκτορα από τον στόχο του. Συγκεκριμένα, αντικαταστήσαμε τα Offset\_x και Offset\_y με την απόσταση του πράκτορα από την ελεύθερη θέση στάθμευσης (*Distance*). Στη συνέχεια, το distance διακριτοποιήθηκε λογαριθμικά, έτσι ώστε να παρέχει μεγαλύτερη ακρίβεια στον πράκτορα, όταν αυτός βρίσκεται κοντά στη θέση στάθμευσης και λιγότερη ακρίβεια, όταν αυτή δεν είναι απαραίτητη, δηλαδή όταν ο πράκτορας απέχει πολύ από τη θέση στάθμευσης. Ωστόσο, τελικά προτιμήθηκε η χρήση των Offset\_x και Offset\_y, καθώς προσφέρουν επιπλέον την πληροφορία της κατεύθυνσης της θέσης, ενώ μειώνουν περισσότερο το μέγεθος του πίνακα Q.

Για την αντιμετώπιση του διλήμματος της εξερεύνησης-αξιοποίησης, χρησιμοποιήσαμε την τεχνική decaying  $\epsilon$ -greedy. Συγκεκριμένα, για τη μείωση της τιμής του  $\epsilon$  χρησιμοποιήσαμε την εξίσωση 5.1:

$$\epsilon = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min}) \times e^{(-\text{decay\_rate} \times \text{episode})} \quad (5.1)$$

όπου  $\epsilon_{\min} = 0.0001$ ,  $\epsilon_{\max} = 1$  και  $\text{decay\_rate} = 0.0001$ . Επιλέξαμε τις τιμές αυτές μετά από διάφορους πειραματισμούς. Σημαντική ήταν η παράμετρος του ρυθμού εξασθένησης του  $\epsilon$  (decay\_rate), καθώς αυτή επηρεάζει τον ρυθμό σύγκλισης του αλγορίθμου. Ωστόσο, καταλήξαμε στην τιμή 0.0001, καθώς με μικρότερες τιμές, ο αλγόριθμος συνέκλινε σημαντικά αργότερα και χρειαζόταν μεγαλύτερος αριθμός βημάτων εκπαίδευσης, αλλά τελικά ο πράκτορας έφτανε στο ίδιο επίπεδο απόδοσης. Από την άλλη, με μεγαλύτερες τιμές, ο αλγόριθμος συνέκλινε γρηγορότερα, αλλά σε χαμηλότερο επίπεδο απόδοσης. Έτσι, η εξασθένηση του  $\epsilon$  φαίνεται στην Εικόνα 5.15.



Εικόνα 5.15. Εξασθένηση του  $\epsilon$ .

## 5.8 Εκπαιδεύσεις με τον αλγόριθμο Q-Learning

---

Από την εμπειρία μας, παρατηρήσαμε πως η απόδοση του πράκτορα δεν βελτιωνόταν περαιτέρω κατά την εκπαίδευση, όταν το  $\epsilon$  προσέγγιζε το 0. Για αυτό, σταματούσαμε τις εκπαιδεύσεις στα 20000 βήματα.

Επίσης, σημαντικό ρόλο παίζει η παράμετρος του ρυθμού μάθησης  $\alpha$  (*learning rate*). Πειραματιστήκαμε με την εξασθένηση κι αυτής της παραμέτρου κατά την εκπαίδευση, ώστε ο πράκτορας να μαθαίνει πιο γρήγορα στην αρχή και να σταθεροποιεί την εκπαίδευσή του στο τέλος. Ωστόσο, καταλήξαμε τελικά στη σταθερή τιμή  $\alpha = 1$ , καθώς το περιβάλλον είναι ντετερμινιστικό κι έτσι, μία συγκεκριμένη ενέργεια σε μία συγκεκριμένη κατάσταση, θα δίνει πάντα την ίδια ανταμοιβή στον πράκτορα.

Κατά τη διάρκεια των εκπαιδεύσεων, δοκιμάστηκαν διάφορες συναρτήσεις ανταμοιβής, όπως και αυτές που είδαμε στην προηγούμενη ενότητα. Όμως, ο πράκτορας δεν κατάφερε να φτάσει με καμία σε ικανοποιητική απόδοση, ενώ συνέκλινε συχνά σε υποβέλτιστες πολιτικές, όπως το να κινείται διαδοχικά εμπρός-πίσω όταν βρίσκεται κοντά στον στόχο του, ώστε να λαμβάνει μικρότερη τιμωρία για την απόσταση του.

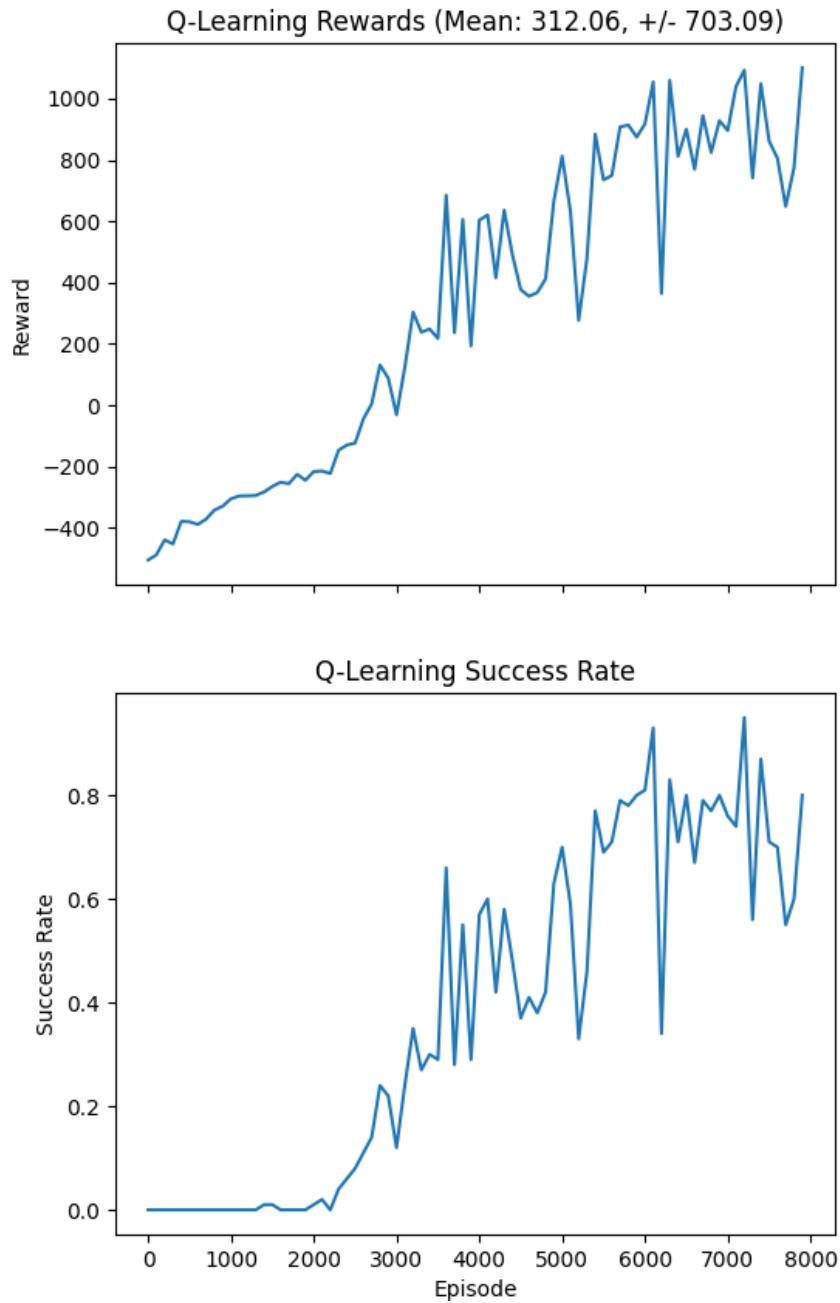
Ακόμα, δοκιμάστηκε η μέθοδος του early stopping, με τα επεισόδια να τερματίζονται στην 1<sup>η</sup> σύγκρουση του πράκτορα με άλλο αντικείμενο. Ωστόσο, αυτή η μέθοδος δεν έφερε καμία βελτίωση στην απόδοση του πράκτορα.

### 5.8.2 Καλύτερες εκπαιδεύσεις

Οι εκπαιδεύσεις με τον αλγόριθμο Q-Learning δεν κατάφεραν να φτάσουν σε ικανοποιητική απόδοση στο συνολικό πρόβλημα, δηλαδή στο επίπεδο δυσκολίας 4. Ωστόσο, σε απλοποιήσεις του προβλήματος, οι πράκτορες κατάφεραν να πετύχουν κάποια, ενδεικτικά αποτελέσματα, τα οποία αποτελούν απόδειξη της σωστής λειτουργίας του αλγορίθμου.

Συγκεκριμένα, στο επίπεδο δυσκολίας 1, όπου τόσο η θέση στάθμευσης, όσο και η αρχική θέση του πράκτορα είναι προκαθορισμένες και η στάθμευση είναι άμεση, ο πράκτορας ανέπτυξε μία ικανή πολιτική, όπως φαίνεται στην Εικόνα 5.16.

Αξιολογώντας τον πράκτορα, παρατηρήσαμε πως παρκάρει επιτυχώς, σε αυτήν τη συγκεκριμένη περίπτωση.

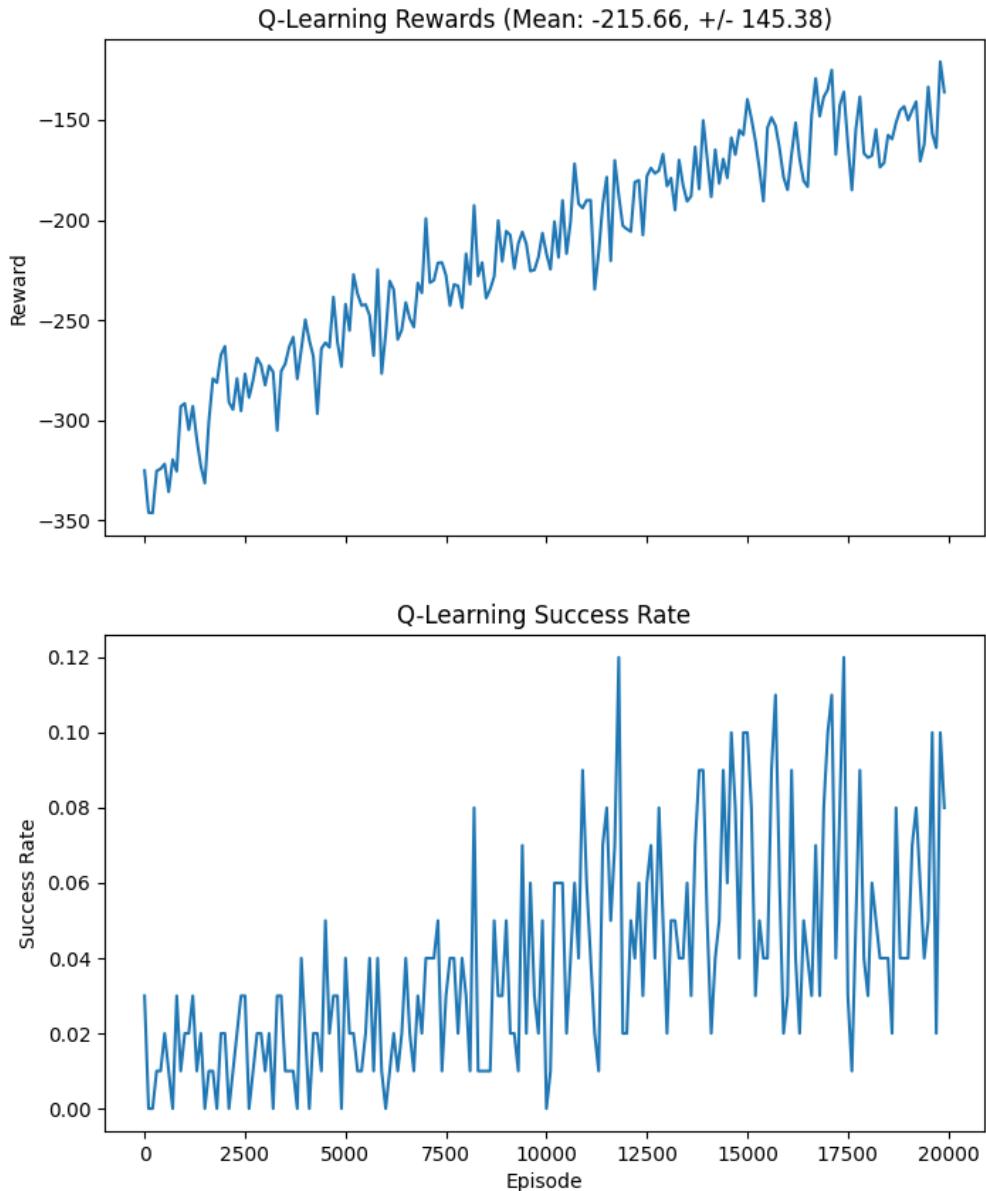


**Εικόνα 5.16.** Καλύτερη εκπαίδευση του Q-Learning στο επίπεδο δυσκολίας 1.

## 5.8 Εκπαίδευσης με τον αλγόριθμο Q-Learning

---

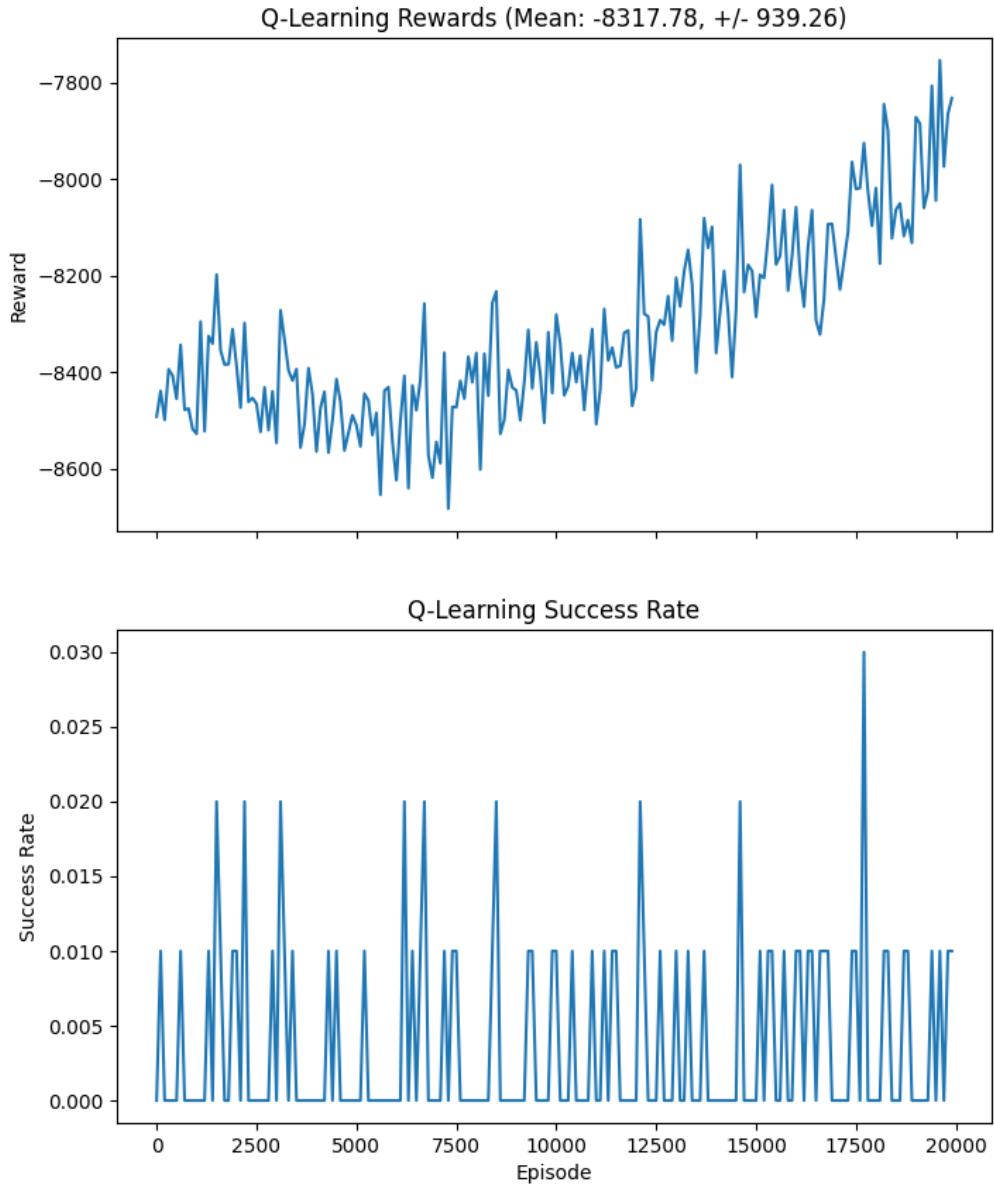
Στο επίπεδο δυσκολίας 2, όπου η θέση στάθμευσης είναι ημι-τυχαία, ο πράκτορας πέτυχε την καλύτερη του απόδοση, με τη συνάρτηση με διαμόρφωση ανταμοιβής (βλ. υποενότητα 5.7.2). Οι μετρικές της εκπαίδευσης φαίνονται στην *Εικόνα 5.17*.



**Εικόνα 5.17.** Καλύτερη εκπαίδευση του Q-Learning στο επίπεδο δυσκολίας 2.

Παρατηρούμε πως η γραφική της μέσης ανταμοιβής έχει την επιθυμή, θετική κλίση, όμως το ποσοστό επιτυχίας κυμαίνεται σε πολύ χαμηλά επίπεδα, μικρότερα του 10%.

Τέλος, στο επίπεδο 3, όπου η θέση στάθμευσης είναι πλήρως τυχαία, τα αποτελέσματα ήταν ακόμα χειρότερα, όπως φαίνεται και στην *Εικόνα 5.18*.



**Εικόνα 5.18.** Καλύτερη εκπαίδευση του Q-Learning στο επίπεδο δυσκολίας 3.

Συγκεκριμένα, βλέπουμε πως το ποσοστό επιτυχίας του πράκτορα είναι σχεδόν μηδαμινό. Πράγματι, κατά την αξιολόγηση του πράκτορα σε 1000 επεισόδια, το ποσοστό επιτυχίας του ήταν 1.1%.

Από τα παραπάνω αποτελέσματα, γίνεται κατανοητό πως η τεχνική της διακριτοποίησης δεν ήταν αρκετή για την επίλυση του προβλήματος. Για αυτό, στη συνεχεία επιλέχθηκε η μέθοδος της προσέγγισης συνάρτησης, δηλαδή η χρήση νευρωνικών δικτύων. Ωστόσο, δεν επιλέξαμε τον αλγόριθμο DQN, αλλά άλλους αλγορίθμους βαθιάς ενισχυτικής μάθησης, καθώς ο DQN δέχεται συνήθως ως είσοδο την εικόνα του παιχνιδιού (*raw pixel data*) κι έτσι, απαιτεί μεγάλη υπολογιστική ισχύ.

## 5.9 Εκπαιδεύσεις με τον αλγόριθμο PPO

### 5.9.1 Επισκόπηση εκπαιδεύσεων

Ο αλγόριθμος PPO ήταν ο πρώτος αλγόριθμος βαθιάς ενισχυτικής μάθησης που εξετάστηκε κι έτσι, δοκιμάστηκαν σε αυτόν πολλές διαφορετικές τεχνικές. Συνολικά, διεξήχθησαν 107 εκπαιδεύσεις του αλγορίθμου, οι οποίες διήρκησαν αθροιστικά 39 ημέρες, 22 ώρες και 56 λεπτά. Ορισμένες από τις μεθόδους που δοκιμάστηκαν, αλλά δεν είχαν τα επιθυμητά αποτελέσματα, αναλύονται παρακάτω.

Αρχικά, δοκιμάστηκαν διάφορες συναρτήσεις ανταμοιβής, όπως αυτές που περιγράφηκαν στην υποενότητα [5.7.2](#), με πολλές παραλλαγές των τιμών των ανταμοιβών τους. Μερικές από τις πιο ενδιαφέρουσες ίδεες που εξετάστηκαν, ήταν οι εξής:

- Αλλαγή της τιμωρίας με βάση την απόσταση πράκτορα-στόχου, σε επιβράβευση, με βάση την ίδια απόσταση. Η ίδεα αυτή βασίστηκε στη θεώρηση, πως η θετική ανταμοιβή μπορεί να ενθαρρύνει τον πράκτορα να εξερευνήσει το περιβάλλον, ενώ η συνεχής αρνητική ανταμοιβή ίσως καθιστά πιο δύσκολη την εξερεύνηση, καθώς ο πράκτορας λαμβάνει συνεχώς ποινές.
- Προσθήκη αυξανόμενης τιμωρίας, με βάση τα βήματα του πράκτορα. Η ίδεα αυτή βασίστηκε στη σκέψη, πως ο πράκτορας μπορεί να μάθει να εκτελεί την εργασία του πιο γρήγορα, αν τιμωρείται για κάθε περιττό βήμα που κάνει.
- Αλλαγή της συνάρτησης ανταμοιβής, ώστε αυτή να προσομοιώσει την διαδικασία στάθμευσης από έναν άνθρωπο. Συγκεκριμένα, ο πράκτορας πρώτα επιβραβεύόταν για την σωστή θέση του στον άξονα x, μετά για τη σωστή γωνία και τέλος, για τη σωστή θέση του στον άξονα y.

Σχετικά με τις παραμέτρους του αλγορίθμου, δοκιμάσαμε διαφορετικές τιμές για τον ρυθμό μάθησης και τον συντελεστή εντροπίας, ενώ σχετικά με το νευρωνικό δίκτυο του αλγορίθμου, δοκιμάσαμε την αλλαγή της συνάρτησης ενεργοποίησης του σε ReLU, καθώς και διαφορετικά πλήθη νευρώνων για τα κρυφά επίπεδα του. Ακόμα, εξετάστηκε η μετατροπή του χώρου ενεργειών σε συνεχή, όπως στην περίπτωση των αλγορίθμων δράστη-κριτή. Ωστόσο, δεν παρατηρήθηκε καμία διαφορά στην απόδοση των πρακτόρων.

Επιπλέον, δοκιμάστηκε η μέθοδος εξερεύνησης generalized State Dependent Exploration (gSDE), αντί της προεπιλεγμένης εξερεύνησης του αλγορίθμου (action noise exploration). Η SDE αποτελεί μια μέθοδο εξερεύνησης, η οποία προσθέτει θόρυβο στην ενέργεια του πράκτορα, με βάση την τρέχουσα κατάσταση. Αυτό έχει ως αποτέλεσμα, μία πιο ομαλή και συνεπή εξερεύνηση, σε σχέση με άλλες μεθόδους. Η gSDE είναι μία βελτίωση της SDE, η οποία προσθέτει τη δειγματοληψία των παραμέτρων της συνάρτησης εξερεύνησης ανά συγκεκριμένο αριθμό βημάτων, καθώς και τη χρήση επιλεγμένων χαρακτηριστικών της πολιτικής, ως είσοδο στη συνάρτηση εξερεύνησης (Schumacher 2023). Δοκιμάσαμε η δειγματοληψία να γίνεται κάθε 1, 4, 20 και 100 βήματα, αλλά σε κάθε περίπτωση, τα αποτελέσματα ήταν χειρότερα σε σχέση με την προεπιλεγμένη μέθοδο εξερεύνησης του αλγορίθμου κι έτσι, η μέθοδος gSDE εγκαταλείφθηκε.

Παρόμοια ήταν τα αποτελέσματα, όταν εξετάστηκε η τεχνική του early stopping στην 2<sup>η</sup> σύγκρουση του πράκτορα. Συγκεκριμένα, αυτή απλώς ώθησε τον πράκτορα να κάνει τον γύρο του χάρτη, αποφεύγοντας έτσι τις συγκρούσεις, αλλά χωρίς να πετυχαίνει τον στόχο της στάθμευσης.

Ακόμα, δοκιμάστηκε η αφαίρεση των episode cutoffs, δηλαδή του ανώτατου ορίου των 600 βημάτων για κάθε επεισόδιο. Τα αποτελέσματα ήταν ενδιαφέροντα, αλλά όχι ικανοποιητικά. Συγκεκριμένα, το μέσο μήκος των επεισοδίων κατά την εκπαίδευση ήταν 2500 βήματα, δηλαδή ο πράκτορας χρειαζόταν περίπου τόσα βήματα για να παρκάρει. Ωστόσο, η συνολική ανταμοιβή κάθε επεισοδίου ήταν αρνητική σε μεγάλο βαθμό, καθώς ο πράκτορας προτού παρκάρει, συγκρουόταν επανειλημμένα με άλλα αντικείμενα.

Τέλος, δοκιμάστηκε η τεχνική του FrameSkip, η οποία εφαρμόστηκε με επιτυχία στους αλγορίθμους δράστη-κρίτη, αλλά δεν είχε ικανοποιητικά αποτελέσματα στον αλγόριθμο PPO κι έτσι, δεν χρησιμοποιήθηκε στις καλύτερες εκπαιδεύσεις του.

## 5.9.2 Καλύτερες εκπαιδεύσεις

Τελικά, τα καλύτερα αποτελέσματα με τον αλγόριθμο PPO, προέκυψαν με την αρχιτεκτονική του δικτύου και με τη συνάρτηση ανταμοιβής που αναλύθηκαν στην ενότητα 5.7, καθώς και με τιμή του συντελεστή εντροπίας ίση με 0.01.

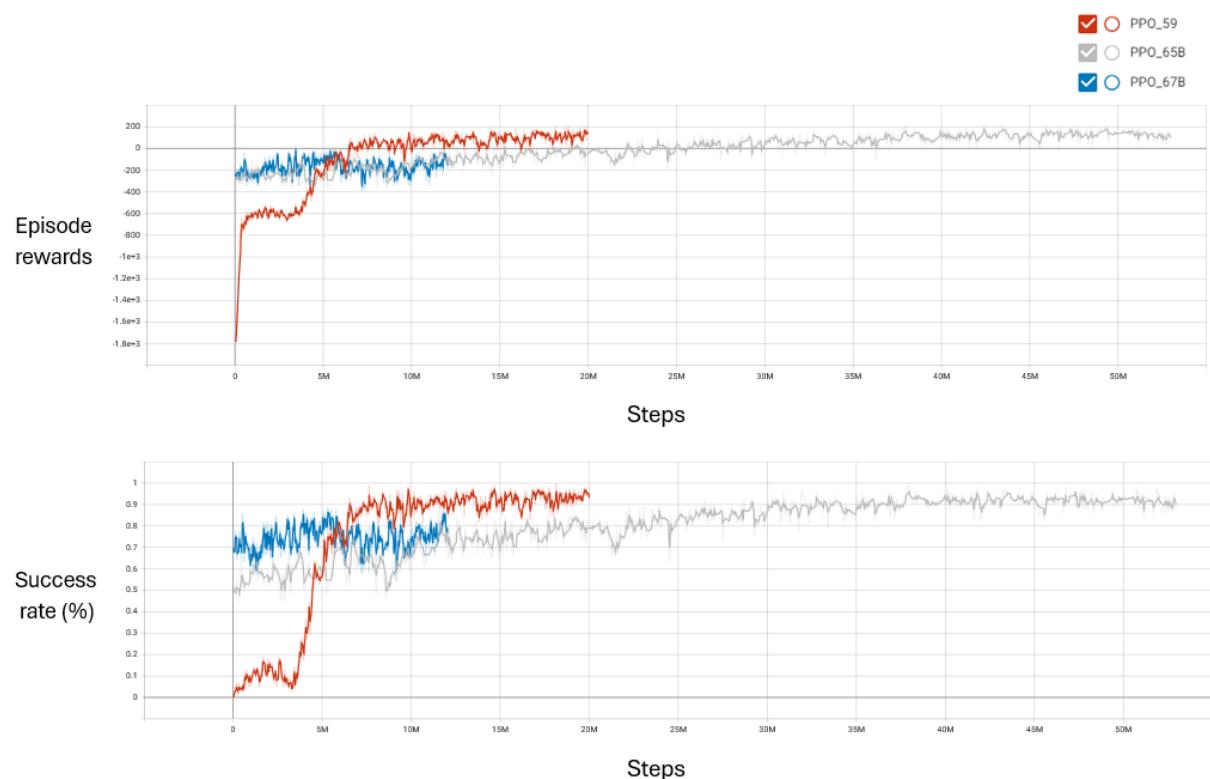
Συγκεκριμένα, στο επίπεδο δυσκολίας 1, μετά από 5M steps, το success rate του πράκτορα προσεγγίζει το 100%. Ωστόσο, με τις ίδιες παραμέτρους εκπαίδευσης, τα αποτελέσματα στο επίπεδο δυσκολίας 2 δεν ήταν ικανοποιητικά. Για αυτό, αναγκαστήκαμε να παρεμβάλουμε ένα ενδιάμεσο επίπεδο δυσκολίας, μεταξύ των 1 και 2, στο οποίο η αρχική θέση του πράκτορα θα είναι τυχαία, αλλά ανάλογα με αυτήν, η ελεύθερη θέση στάθμευσης θα είναι είτε η κεντρική θέση της πάνω σειράς θέσεων, είτε της κάτω. Στο επίπεδο αυτό, μετά από 8M steps, το success rate του πράκτορα προσεγγίζει το 100%, όπως φαίνεται από την κόκκινη καμπύλη στην Εικόνα 5.19 (εκπαίδευση 59).

Στη συνέχεια, εφαρμόσαμε Curriculum Learning στον πράκτορα των 20M steps της εκπαίδευσης 59,

## 5.9 Εκπαίδευσης με τον αλγόριθμο PPO

εκπαιδεύοντας τον στο επίπεδο δυσκολίας 2. Μετά από 53M steps, το success rate συγκλίνει στο 95%, όπως φαίνεται από τη γκρι καμπύλη στην Εικόνα 5.19 (εκπαίδευση 65B). Παρατηρούμε μάλιστα, πως το success rate ξεκινάει από την τιμή 50%, το οποίο είναι αναμενόμενο, λόγω της εφαρμογής κλιμακωτής μάθησης. Με άλλα λόγια, ήδη από την αρχή της εκπαίδευσης, ο πράκτορας έχει κάποια χρήσιμη γνώση για να λύσει το πρόβλημα.

Η ίδια τεχνική εφαρμόστηκε και για το επίπεδο 3, επανεκπαιδεύοντας αυτή τη φορά τον πράκτορα των 50M steps της εκπαίδευσης 65B. Μετά από 12M steps, το success rate κυμαίνεται γύρω από το 75%, όπως φαίνεται από τη μπλε καμπύλη στην Εικόνα 5.19 (εκπαίδευση 67B). Ωστόσο, η εκπαίδευση 67B δεν συνεχίστηκε περαιτέρω, λόγω χρονικών περιορισμών.



**Εικόνα 5.19.** Καλύτερες εκπαίδευσης του αλγορίθμου PPO: με κόκκινο χρώμα στο επίπεδο μεταξύ 1 και 2, με γκρι χρώμα στο επίπεδο 2 και με μπλε χρώμα στο επίπεδο 3.

Παρόλα αυτά, όταν δοκιμάστηκε η εφαρμογή Curriculum Learning για το επίπεδο δυσκολίας 4, τα αποτελέσματα δεν ήταν αντίστοιχα με πριν. Αντίθετα, μετά από 40M steps, το success rate συνέκλινε στην τιμή 20%. Επομένως, οι εκπαίδευσης με τον αλγόριθμο PPO απέτυχαν να λύσουν το πραγματικό πρόβλημα της αυτόματης στάθμευσης.

## 5.10 Εκπαιδεύσεις με τον αλγόριθμο SAC

### 5.10.1 Επισκόπηση εκπαιδεύσεων

Ο αλγόριθμος SAC ήταν ο πρώτος της κατηγορίας δράστη-κριτή που εξετάστηκε. Συνολικά, διεξήχθησαν 21 εκπαιδεύσεις του αλγορίθμου, οι οποίες διήρκησαν αθροιστικά 26 ημέρες, 13 ώρες και 26 λεπτά.

Μία ενδιαφέρουσα συμπεριφορά, που προέκυψε κατά τις εκπαιδεύσεις των πρακτόρων με τον αλγόριθμο SAC, ήταν η αδυναμία τους να μάθουν να παρκάρουν κινούμενο προς τα εμπρός. Συγκεκριμένα, παρατηρήσαμε πως όταν στην αρχική θέση του αυτοκινήτου, αυτό ήταν στραμμένο με την πίσω μεριά του προς τη θέση στάθμευσης, τότε ο πράκτορας πάρκαρε πάντα με την όπισθεν. Αντίθετα, όταν στην αρχική θέση του αυτοκινήτου, αυτό ήταν στραμμένο με την μπροστινή μεριά του προς τη θέση στάθμευσης ή έστω παράλληλα προς αυτήν, τότε ο πράκτορας έμενε ακίνητος. Προκειμένου να αντιμετωπίστει αυτό το προβλήμα, προσθέσαμε στη συνάρτηση ανταμοιβής επιβράβευση, για την κίνηση του πράκτορα προς τα εμπρός. Πράγματι, με τον τρόπο αυτό, ο πράκτορας έμαθε να παρκάρει τόσο με την όπισθεν, όσο και προς τα εμπρός.

Ακόμα, αξίζει να γίνει αναφορά στη χρησιμότητα της τεχνικής FrameSkip, καθώς αυτή αποτέλεσε το κλειδί για την επιτυχία του πράκτορα, στα μεγαλύτερα επίπεδα δυσκολίας του παιχνιδιού. Συγκεκριμένα, εφαρμόσαμε FrameSkip 4 βημάτων και χρειάστηκε επίσης, να μεταβαλούμε σε 4 την παράμετρο `train_freq` της βιβλιοθήκης Stable-Baselines3, η οποία ορίζει το πλήθος των βημάτων εκπαίδευσης, που θα γίνουν πριν από κάθε ενημέρωση του πράκτορα.

Τέλος, κρίσιμη αποδείχθηκε η χρήση της τεχνικής Curriculum Learning, προκειμένου να επιτευχθούν καλά αποτελέσματα από τον αλγόριθμο, στο τελευταίο επίπεδο δυσκολίας του παιχνιδιού. Συγκεκριμένα, χωρίς την τεχνική αυτή, ο πράκτορας υιοθετούσε απλώς την υποβέλτιστη πολιτική του να κάνει τον γύρο του χάρτη.

### 5.10.2 Καλύτερες εκπαιδεύσεις

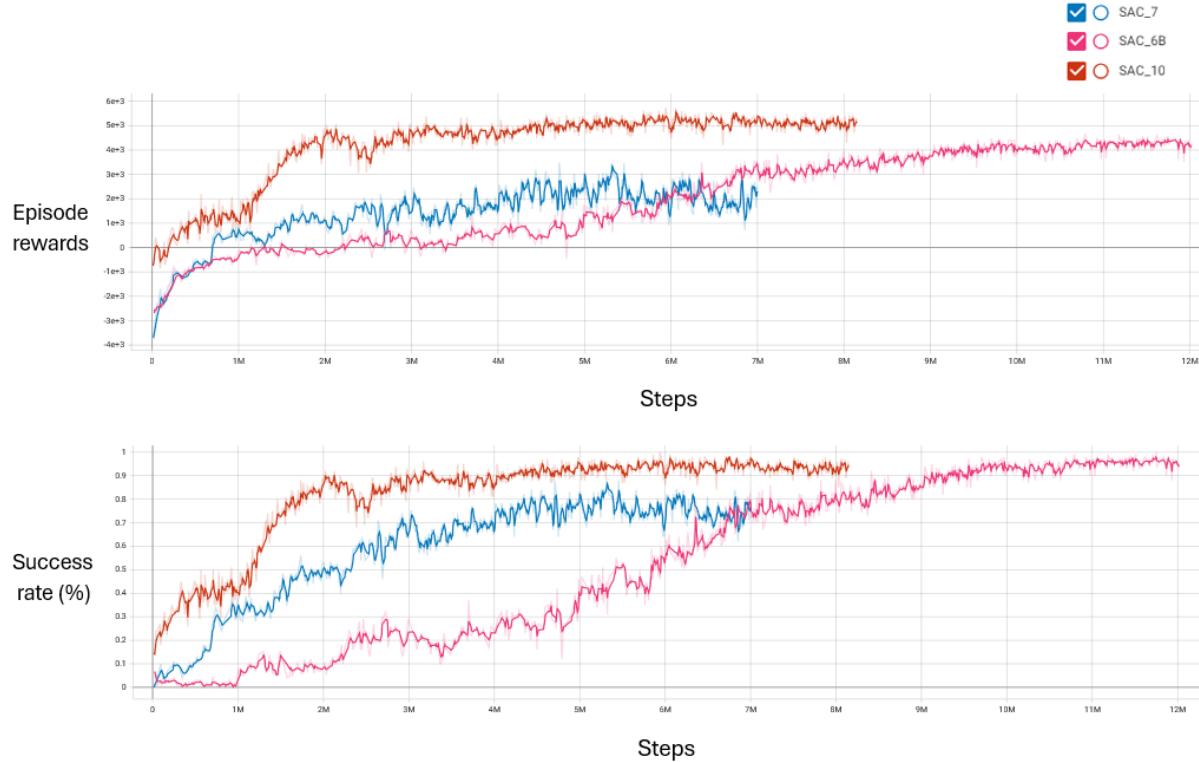
Οι εκπαιδεύσεις ξεκίνησαν από το επίπεδο δυσκολίας 2, του οποίου η καλύτερη εκπαίδευση με τον αλγόριθμο SAC διακρίνεται στην Εικόνα 5.20 με μπλε χρώμα (εκπαίδευση 7). Παρατηρούμε πως μετά από 7M steps, η γραφική του success rate συγκλίνει στο 80%.

Στη συνέχεια, χρησιμποιήθηκαν οι ίδιες παράμετροι εκπαίδευσης για το επίπεδο δυσκολίας 3. Η καλύτερη εκπαίδευση σε αυτό το επίπεδο φαίνεται στην Εικόνα 5.20 με ροζ χρώμα (εκπαίδευση 6B). Παρατηρούμε πως μετά από 12M steps, το success rate προσεγγίζει το 100%.

Τέλος, για να πετύχει ο αλγόριθμος ικανοποιητικές επιδόσεις στο επίπεδο δυσκολίας 4, χρειάστηκε να εφαρμόσουμε Curriculum Learning με τον πράκτορα των 12M steps της εκπαίδευσης 6B. Η εκπαίδευση

## 5.11 Εκπαιδεύσεις με τον αλγόριθμο TD3

αυτή εμφανίζεται στην *Εικόνα 5.20* με κόκκινο χρώμα (εκπαίδευση 10). Παρατηρούμε πως μετά από 8M steps, το success rate συγκλίνει στο 95%.



**Εικόνα 5.20.** Καλύτερες εκπαιδεύσεις του αλγορίθμου SAC: με μπλε χρώμα στο επίπεδο 2, με ροζ χρώμα στο επίπεδο 3 και με κόκκινο χρώμα στο επίπεδο 4.

Επομένως, βλέπουμε πως οι πράκτορες του αλγορίθμου SAC έφτασαν σε υψηλά επίπεδα επιδόσεων, σε όλα τα επίπεδα δυσκολίας και άρα, πέτυχαν την επίλυση του προβλήματος της αυτόματης στάθμευσης.

## 5.11 Εκπαιδεύσεις με τον αλγόριθμο TD3

### 5.11.1 Επισκόπηση εκπαιδεύσεων

Ο αλγόριθμος TD3 ήταν ο δεύτερος της κατηγορίας δράστη-κριτή που εξετάστηκε και αυτός που εν τέλει, πέτυχε τα καλύτερα αποτελέσματα. Συνολικά, διεξήχθησαν 32 εκπαιδεύσεις του αλγορίθμου, οι οποίες διήρκησαν αθροιστικά 21 ημέρες, 23 ώρες και 47 λεπτά.

Αρχικά, δοκιμάσαμε την εκπαίδευση με χρήση της αραιής συνάρτησης ανταμοιβής της υποενότητας [5.7.2](#). Ωστόσο, ο πράκτορας υιοθέτησε την υποβέλτιστη πολιτική του να στέκεται ακίνητος καθ' όλη

τη διάρκεια του επεισοδίου. Έτσι, δοκιμάσαμε τη συνάρτηση με διαμόρφωση ανταμοιβής και μέσω αυτής, επιτεύχθηκαν γρήγορα, ικανοποιητικά αποτελέσματα. Επομένως, μπορέσαμε να αφιερώσουμε χρόνο στη βελτίωση της πολιτικής του πράκτορα, η οποία πάσχιζε από τα εξής δύο προβλήματα:

- Συχνές συγκρούσεις: για παράδειγμα, σε μία εκπαίδευση, ο πράκτορας έμαθε να παρκάρει επιτυχώς και με συνέπεια, όμως κάθε φορά, συγκρουόταν πρώτα με τον κήπο του χάρτη.
- Εξισορρόπηση μεταξύ στάθμευσης προς τα εμπρός και με την όπισθεν: σε κάποιες εκπαίδεύσεις, ο πράκτορας είχε την τάση να παρκάρει πάντα με την όπισθεν, ενώ σε άλλες γινόταν το ανάποδο.

Τα παραπάνω προβλήματα ατιμετωπίστηκαν επιτυχώς, μετά από πολλαπλές τροποποιήσεις των τιμών των ανταμοιβών του πράκτορα.

Τέλος, όπως και στην περίπτωση του SAC, οφείλουμε να σημειώσουμε την αξία της τεχνικής Frameskip, η οποία αποτέλεσε τον παράγοντα που οδήγησε στην επιτυχία του πράκτορα, στο επίπεδο δυσκολίας 3 (αυτόματη στάθμευση).

Αντίστοιχα, στο επίπεδο δυσκολίας 4 (κανονική στάθμευση), ο παράγοντας που αποδείχθηκε καθοριστικός ήταν η κλιμακωτή μάθηση, καθώς χωρίς αυτήν, το ποσοστό επιτυχίας του πράκτορα ήταν μηδαμινό.

### 5.11.2 Καλύτερες εκπαίδεύσεις

Οι εκπαίδεύσεις ξεκίνησαν από το επίπεδο δυσκολίας 2, του οποίου η καλύτερη εκπαίδευση με τον αλγόριθμο TD3 διακρίνεται στην Εικόνα 5.21 με κόκκινο χρώμα (εκπαίδευση 5C). Παρατηρούμε πως μετά από 4M steps, η γραφική του success rate προσεγγίζει το 100%.

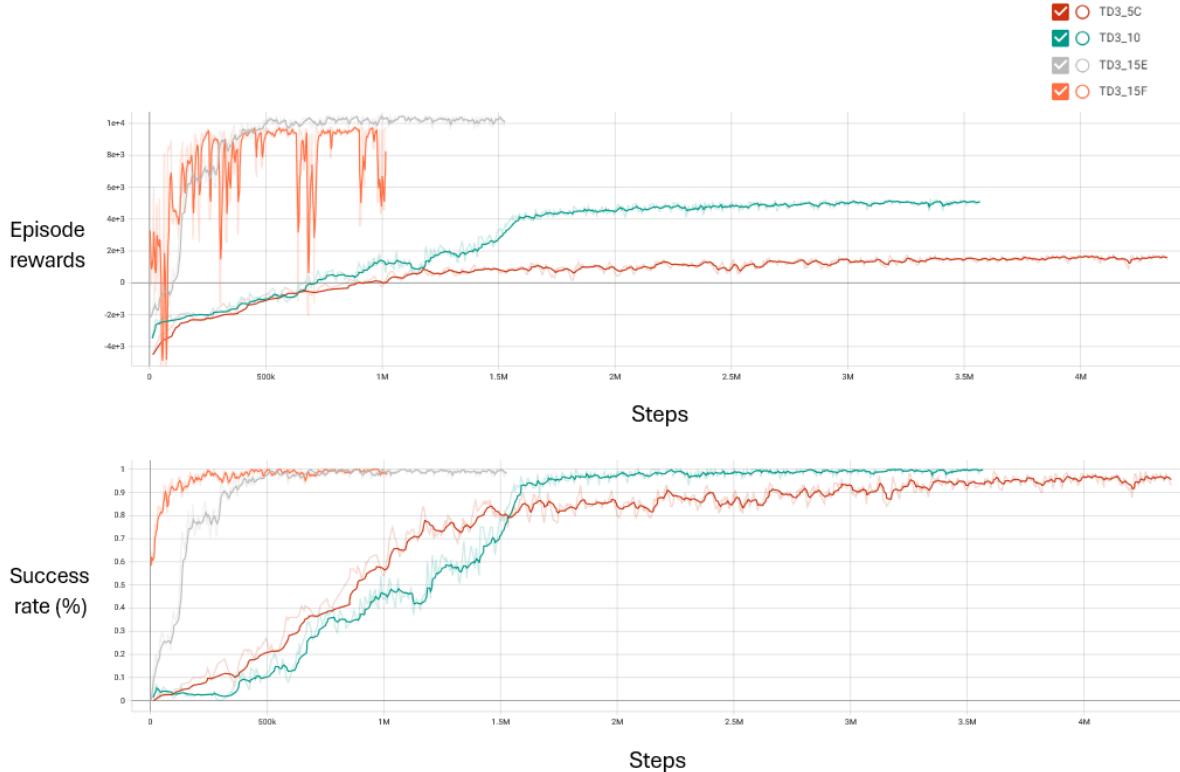
Στη συνέχεια, χρησιμοποιήθηκαν οι ίδιες παράμετροι εκπαίδευσης για το επίπεδο δυσκολίας 3. Η καλύτερη εκπαίδευση σε αυτό το επίπεδο φαίνεται στην Εικόνα 5.21 με τιρκουάζ χρώμα (εκπαίδευση 10). Παρατηρούμε πως μετά από 2M steps, το success rate προσεγγίζει το 100%.

Έπειτα, για να πετύχει ο αλγόριθμος ικανοποιητικές επιδόσεις στο επίπεδο δυσκολίας 4, χρειάστηκε να εφαρμόσουμε Curriculum Learning με τον πράκτορα των 3M steps της εκπαίδευσης 10. Η εκπαίδευση αυτή εμφανίζεται στην Εικόνα 5.21 με γκρι χρώμα (εκπαίδευση 15E). Παρατηρούμε πως μετά από μόλις 400K steps, το success rate προσεγγίζει το 100%.

Τέλος, εφαρμόσαμε ξανά την τεχνική Curriculum Learning, αυτή τη φορά με τον πράκτορα των 1.5M steps της εκπαίδευσης 15E, αυξάνοντας την τιμωρία για τις συγκρούσεις, προκειμένου να μειώσουμε τη συχνότητα αυτών. Η εκπαίδευση αυτή εμφανίζεται στην Εικόνα 5.21 με πορτοκαλί χρώμα (εκπαίδευση 15F). Παρατηρούμε πως μετά από 300K steps, το success rate προσεγγίζει το 100%. Η γραφική των ανταμοιβών του πράκτορα συγκλίνει ξανά στην ίδια τιμή με πριν και άρα, καταλαβαίνουμε πως οι συγκρούσεις έχουν μειωθεί. Ωστόσο, στη γραφική εμφανίζονται πολλές

## 5.12 Εκπαίδευσης με τον αλγόριθμο DDPG

απότομες βυθίσεις, εξαιτίας της μεγάλης πλέον, τιμής της τιμωρίας για τις συγκρούσεις. Επομένως, γίνεται κατανοητό πως οι συγκρούσεις δεν έχουν εξαλειφθεί πλήρως.



**Εικόνα 5.21.** Καλύτερες εκπαίδευσης του αλγορίθμου TD3: με κόκκινο χρώμα στο επίπεδο 2, με τιρκουάζ χρώμα στο επίπεδο 3, με γκρι χρώμα στο επίπεδο 4 και με πορτοκαλί χρώμα στο επίπεδο 4 με αύξηση της τιμωρίας για τις συγκρούσεις.

Επομένως, βλέπουμε πως οι πράκτορες του αλγορίθμου TD3 έφτασαν σε υψηλά επίπεδα επιδόσεων, σε όλα τα επίπεδα δυσκολίας και άρα, πέτυχαν την επίλυση του προβλήματος της αυτόματης στάθμευσης.

## 5.12 Εκπαίδευσης με τον αλγόριθμο DDPG

### 5.12.1 Επισκόπηση εκπαίδευσεων

Ο αλγόριθμος DDPG ήταν ο τελευταίος που εξετάστηκε. Συνολικά, διεξήχθησαν 15 εκπαίδευσης του αλγορίθμου, οι οποίες διήρκησαν αθροιστικά 7 ημέρες, 3 ώρες και 48 λεπτά.

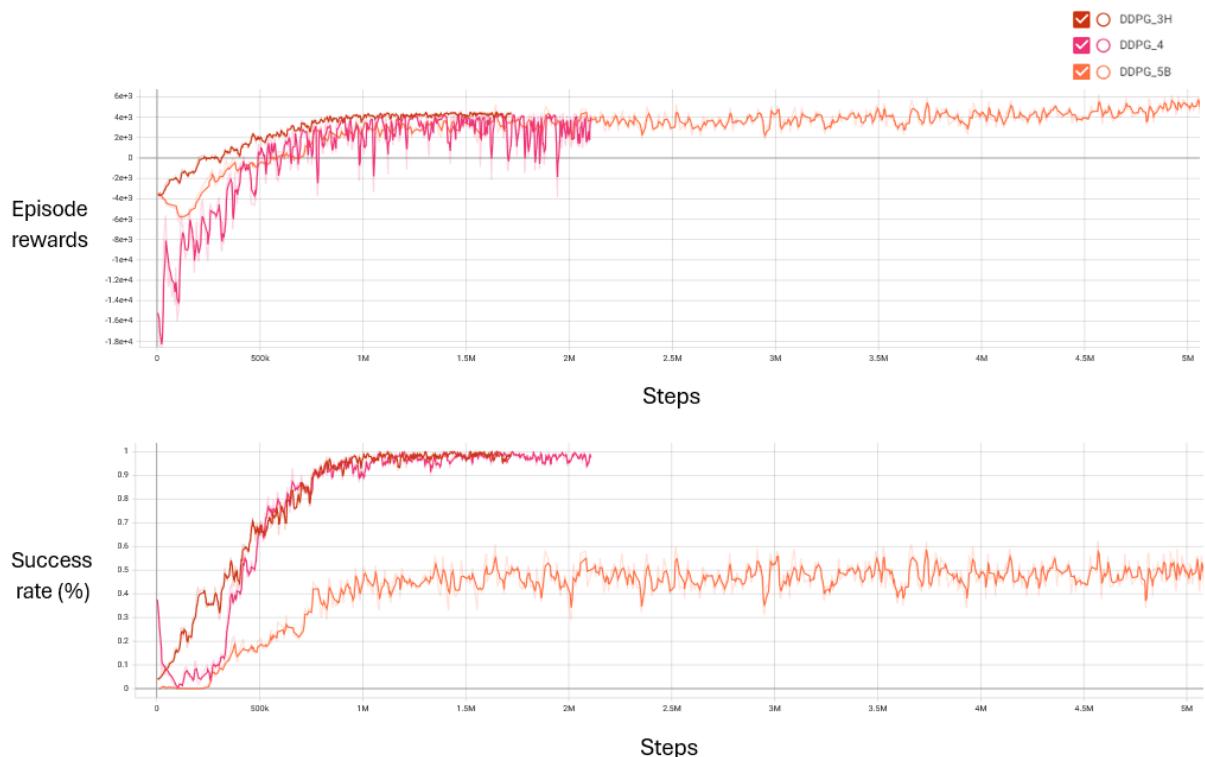
Το μικρό, σχετικά, πλήθος των εκπαίδευσεων, οφείλεται στο γεγονός πως οι παράμετροι που χρησιμοποιήθηκαν για τις εκπαίδευσης του αλγορίθμου DDPG, προέκυψαν από τις προηγούμενες

εκπαίδεύσεις των αλγορίθμων SAC και TD3. Επομένως, δεν απαιτήθηκαν πολλές τροποποιήσεις, προκειμένου να φτάσει ο αλγόριθμος DDPG σε ικανοποιητικά αποτελέσματα.

Οι μοναδικές αλλαγές που έγιναν, αφορούν τη συνάρτηση ανταμοιβής και ειδικότερα, τις ανταμοιβές για την κίνηση του πράκτορα. Συγκεκριμένα, όπως και οι άλλοι αλγόριθμοι της κατηγορίας δράστη-κριτή, οι πράκτορες του DDPG δυσκολεύτηκαν να μάθουν να παρκάρουν και προς τις δύο κατευθύνσεις. Όμως μετά από κατάλληλες μεταβολές των ανταμοιβών, το πρόβλημα αυτό λύθηκε.

### 5.12.2 Καλύτερες εκπαίδεύσεις

Οι εκπαίδεύσεις ξεκίνησαν κατευθείαν από το επίπεδο δυσκολίας 3, καθώς χρησιμοποιήθηκε η ίδια συνάρτηση ανταμοιβής με τον αλγόριθμο TD3 και οι δύο αλγόριθμοι είναι αρκετά παρόμοιοι. Η καλύτερη εκπαίδευση στο επίπεδο αυτό, φαίνεται στην Εικόνα 5.22 με κόκκινο χρώμα (εκπαίδευση 3H). Παρατηρούμε πως μετά από 1.5M steps, το success rate προσεγγίζει το 100%.



**Εικόνα 5.22.** Καλύτερες εκπαίδεύσεις του αλγορίθμου DDPG: με κόκκινο χρώμα στο επίπεδο 3, με ροζ χρώμα στο επίπεδο 3 με αύξηση της τιμωρίας για τις συγκρούσεις και με πορτοκαλί χρώμα στο επίπεδο 4.

Στη συνέχεια, εφαρμόστηκε Curriculum Learning στον πράκτορα των 1.3M steps της εκπαίδευσης 3H, στο ίδιο επίπεδο, αυξάνοντας την τιμωρία για τις συγκρούσεις. Η εκπαίδευση αυτή εμφανίζεται

στην *Εικόνα 5.22* με ροζ χρώμα (εκπαίδευση 4). Παρατηρούμε πως μετά από 1M steps, το success rate προσεγγίζει το 100%. Η γραφική των ανταμοιβών του πράκτορα συγκλίνει ξανά στην ίδια τιμή με πριν και άρα, καταλαβαίνουμε πως οι συγκρούσεις έχουν μειωθεί. Ωστόσο, στη γραφική εμφανίζονται πολλές απότομες βυθίσεις κι έτσι, γίνεται κατανοητό πως οι συγκρούσεις δεν έχουν εξαλειφθεί πλήρως.

Τέλος, εφαρμόστηκε Curriculum Learning με τον πράκτορα των 1.55M steps της εκπαίδευσης 4, στο επίπεδο δυσκολίας 4. Η εκπαίδευση αυτή εμφανίζεται στην *Εικόνα 5.21* με πορτοκαλί χρώμα (εκπαίδευση 5B). Παρατηρούμε πως μετά από μόλις 5M steps, το success rate προσεγγίζει το 50%, ενώ η γραφική των ανταμοιβών του πράκτορα ήδη συγκλίνει (η τιμή της δεν μπορεί να συγκριθεί με τις υπόλοιπες, καθώς σε αυτήν την εκπαίδευση αυξήθηκε η ανταμοιβή της στάθμευσης). Εξετάζοντας τον πράκτορα στο παιχνίδι, παρατηρούμε πως η πολιτική του είναι ασταθής. Συγκεκριμένα, σε κάποια επεισόδια παρκάρει άψογα, ενώ σε άλλα κινείται ακανόνιστα, χωρίς να προσπαθεί να παρκάρει. Σε κάθε περίπτωση, οι προσπάθειες μας να ενθαρρύνουμε τον πράκτορα να παρκάρει περισσότερο συχνά, απέβησαν άκαρπες.

Επομένως, βλέπουμε πως οι πράκτορες του αλγορίθμου DDPG κατάφεραν να πετύχουν καλά αποτελέσματα, εκτός του τελικού επιπέδου δυσκολίας, στο οποίο η απόδοση τους δεν είναι συνεπής και άρα, δεν θεωρείται πως έλυσαν αξιόπιστα το πρόβλημα της αυτόματης στάθμευσης.

## 6 Αξιολόγηση αποτελεσμάτων

Σε αυτό το κεφάλαιο, θα παρούσιασουμε τα αποτελέσματα από τις εκπαιδεύσεις των αλγορίθμων, με πιο οργανωμένο τρόπο, προκειμένου να προβούμε στη σύγκριση τους.

Αρχικά, ας ορίσουμε τη σημασία της σύγκρισης αλγορίθμων ενισχυτικής μάθησης. Προφανώς, το κάθε περιβάλλον εκπαίδευσης έχει τα δικά του χαρακτηριστικά και ιδιαιτερότητες. Επομένως, δεν μπορούμε να γενικεύσουμε τα συμπεράσματα της εργασίας μας, σε όλα τα διαφορετικά περιβάλλοντα. Ωστόσο, η σύγκριση των αλγορίθμων θα είναι έγκυρη, για περιβάλλοντα με παρόμοιες ιδιότητες με αυτό της αυτόματης στάθμευσης. Επομένως, όπως αναφέρεται και στο (Patterson κ.ά. 2023), συγκρίνοντας αλγορίθμους ενισχυτικής μάθησης κάνουμε τον ισχυρισμό: «αν το πρόβλημά σας είναι παρόμοιο με το δικό μας, τότε αυτός είναι ο αλγόριθμος που πρέπει να χρησιμοποιήσετε».

Ακόμα, μία καλή πρακτική κατά την εξέταση της απόδοσης των πρακτόρων, είναι η ποσοτική αξιολόγηση (Raffin 2021). Συγκεκριμένα, είναι καλό να χρησιμοποιείται μεγάλο πλήθος επεισοδίων αξιολόγησης, ώστε τα αποτελέσματα αυτής να θεωρούνται αξιόπιστα και η απόδοση των πρακτόρων σταθερή. Επομένως, στην εργασία μας, χρησιμοποιήσαμε 100 επεισόδια αξιολόγησης, τα οποία ήταν σταθερά για όλους τους αλγορίθμους.

Στις επόμενες ενότητες, θα πραγματοποιήσουμε τη σύγκριση των αλγορίθμων, πρώτα όσον αφορά τον χρόνο εκπαίδευσης (Ενότητα 6.1) και στη συνέχεια όσον αφορά την απόδοση τους (Ενότητα 6.2). Τέλος, θα παρουσιάσουμε τα τελικά συμπεράσματα από την εργασία μας (Ενότητα 6.3).

### 6.1 Σύγκριση αλγορίθμων ως προς τον χρόνο εκπαίδευσης

Ο χρόνος εκπαίδευσης είναι μία ιδιαίτερα χρήσιμη μετρική της αποδοτικότητας (*efficiency*) των αλγορίθμων, καθώς αποτελεί ένδειξη του υπολογιστικού κόστους τους.

Αρχικά, αξίζει να διευκρίνισουμε γιατί επιλέγουμε τη μετρική του χρόνου εκπαίδευσης, έναντι των βημάτων εκπαίδευσης. Ο λόγος είναι, πως η πολυπλοκότητα κάθε αλγορίθμου μεταβάλλει τον χρόνο που απαιτεί ένα βήμα της εκπαίδευσης. Με άλλα λόγια, στον ίδιο αριθμό βημάτων, δύο διαφορετικοί αλγόριθμοι, μπορεί να έχουν εκπαιδευτεί για διαφορετικό χρονικό διάστημα. Συνεπώς, δεν είναι δίκαιο να συγκρίνουμε τον αριθμό των βημάτων μεταξύ των αλγορίθμων, καθώς δεν αποτελεί αντικειμενικό μέτρο. Αντίθετα, ο χρόνος εκπαίδευσης είναι μία πιο αντικειμενική μετρική, καθώς εξαρτάται από την

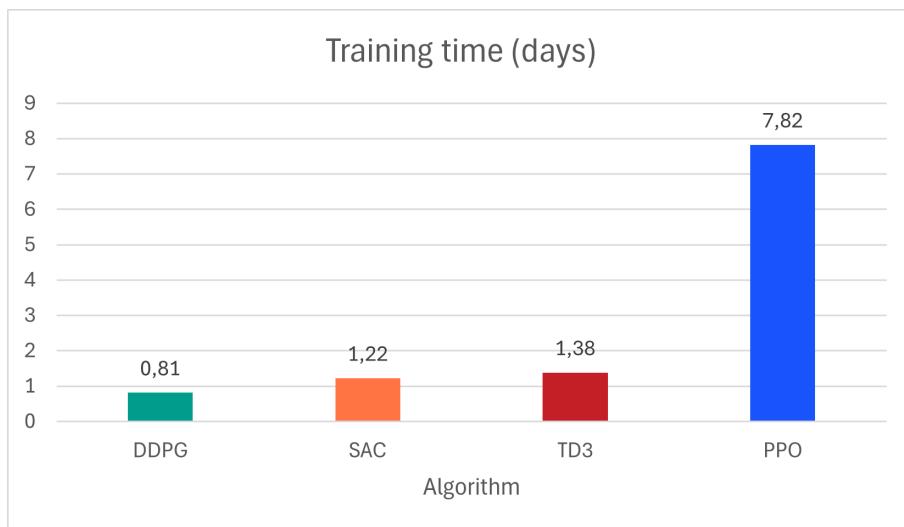
## 6.1 Σύγκριση αλγορίθμων ως προς τον χρόνο εκπαίδευσης

υπολογιστική ισχύ του συστήματος, την πολυπλοκότητα του προβλήματος και την αποδοτικότητα του αλγορίθμου.

Επομένως, αν διατηρήσουμε σταθερές τις πρώτες δύο παραμέτρους, μπορούμε να εκτιμήσουμε με ασφάλεια την αποδοτικότητα των αλγορίθμων. Ωστόσο, στα πλαίσια της εργασίας, χρησιμοποιηθήκαν διαφορετικά μηχανήματα για τις εκπαίδευσις των πρακτόρων, όπως αναλύθηκε στην *Ενότητα 5.3*. Για αυτό, στα γραφήματα που ακολουθούν, κανονικοποιήσαμε τον χρόνο κάθε εκπαίδευσης στον αντίστοιχο που θα απαιτούνταν, εφόσον χρησιμοποιούταν η GPU: NVIDIA GeForce MX110. Με τον τρόπο αυτό, είμαστε σε θέση να διεξάγουμε μία δίκαιη σύγκριση των χρόνων εκπαίδευσης.

### 6.1.1 Επίπεδο δυσκολίας 3 - Άμεση στάθμευση

Στην *Εικόνα 6.1*, παρουσιάζονται οι χρόνοι εκπαίδευσης των καλύτερων πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3, σε ημέρες. Στην εικόνα περιλαμβάνονται μόνο οι αλγόριθμοι που πέτυχαν ικανοποιητικά αποτελέσματα στο επίπεδο αυτό.



**Εικόνα 6.1.** Χρόνοι εκπαίδευσης των καλύτερων πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3.

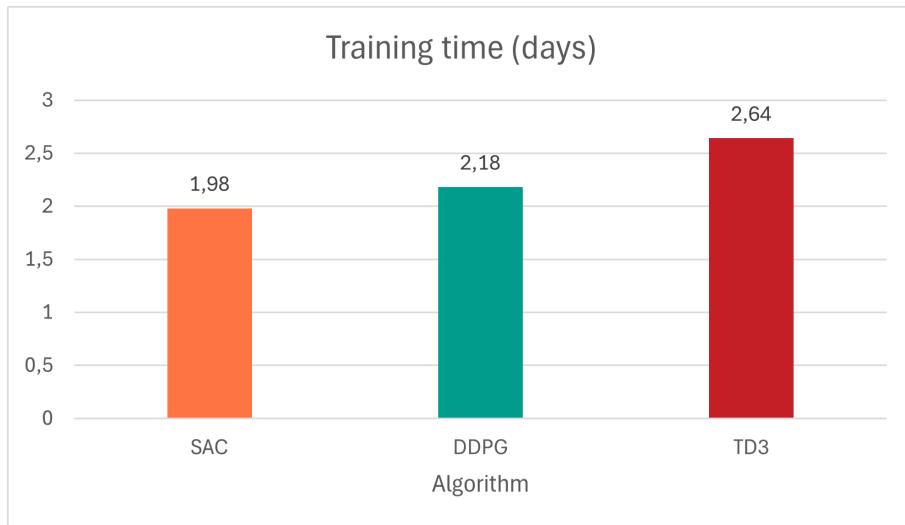
Παρατηρούμε πως οι αλγόριθμοι δράστη-κριτή χρειάστηκαν σημαντικά λιγότερο χρόνο για την εκπαίδευση τους, σε σχέση με τον αλγόριθμο βελτιστοποίησης πολιτικής PPO. Αυτό βέβαια, μάλλον οφείλεται στην πολυπλοκότητα του προβλήματος αυτόματης στάθμευσης, η οποία παρέτεινε την εκπαίδευση του αλγορίθμου PPO, και όχι στο υπολογιστικό κόστος κάθε αλγορίθμου.

Ακόμα, εξετάζοντας τους αλγορίθμους δράστη-κριτή μεταξύ τους, παρατηρούμε πως ο αλγορίθμος DDPG χρειάστηκε τον μικρότερο χρόνο, κάτι που είναι λογικό, καθώς είναι ο πιο απλός αλγόριθμος

της κατηγορίας. Αντίθετα, ο αλγόριθμος TD3 χρειάστηκε τον μεγαλύτερο χρόνο, κάτι που επιβεβαιώνει τον αυξημένο υπολογιστικό φόρτο του αλγορίθμου.

### 6.1.2 Επίπεδο δυσκολίας 4 - Κανονική στάθμευση

Στην *Εικόνα 6.2*, παρουσιάζονται οι χρόνοι εκπαίδευσης των καλύτερων πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 4, σε ημέρες. Στην εικόνα περιλαμβάνονται μόνο οι αλγόριθμοι που πέτυχαν ικανοποιητικά αποτελέσματα στο επίπεδο αυτό και για αυτό, ο αλγόριθμος PPO απουσιάζει.



**Εικόνα 6.2.** Χρόνοι εκπαίδευσης των καλύτερων πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 4.

Βλέπουμε πως η σχέση μεταξύ των αλγορίθμων DDPG και TD3 είναι ίδια με πριν, κι επομένως επαληθεύονται οι παρατηρήσεις που έγιναν προηγουμένως. Ωστόσο, είναι αξιοσημείωτο πως στην προκειμένη περίπτωση, ο αλγόριθμος SAC χρειάστηκε τον μικρότερο χρόνο για την επιτυχή εκπαίδευση του πράκτορα.

## 6.2 Σύγκριση αλγορίθμων ως προς την απόδοση

Στη συγκεκριμένη ενότητα, θα συγκρίνουμε τους αλγορίθμους ως προς την απόδοση τους, δηλαδή ως προς την ικανότητα των πρακτόρων να παρκάρουν. Για να το πετύχουμε αυτό, θα χρησιμοποιήσουμε τις εξής μετρικές:

- **Ποσοστό επιτυχίας:** αποτελεί το προφανές και βασικότερο μέτρο σύγκρισης, της απόδοσης των πρακτόρων.

## 6.2 Σύγκριση αλγορίθμων ως προς την απόδοση

---

- **Μέσος αριθμός συγκρούσεων:** ο μέσος αριθμός συγκρούσεων που προκαλεί ο πράκτορας μέχρι να πετύχει το στόχο του, αποτελεί άλλον έναν τρόπο αξιολόγησης της απόδοσης του.
- **Μέσος χρόνος ολοκλήρωσης:** ο μέσος χρόνος που χρειάζεται ο πράκτορας για να ολοκληρώσει το στόχο του είναι μία ακόμα μετρική της ποιότητας της πολιτικής του.

Ωστόσο, επιθυμούμε να συνοψίσουμε τις τρεις αυτές μετρικές, σε ένα ενιαίο σκορ, το οποίο θα υποδηλώνει, πόσο ικανός είναι κάποιος στο παρκάρισμα (τουλάχιστον στα πλαίσια του παιχνιδιού μας). Το τελικό αυτό σκορ θα υπολογίζεται από την εξίσωση 6.1:

$$\text{score} = 70 \times \text{success\_rate} + \max(20 - 5 \times \text{mean\_collisions}, 0) + 10 \times \max\left(1 - \frac{\text{mean\_time} - 10}{10}, 0\right) \quad (6.1)$$

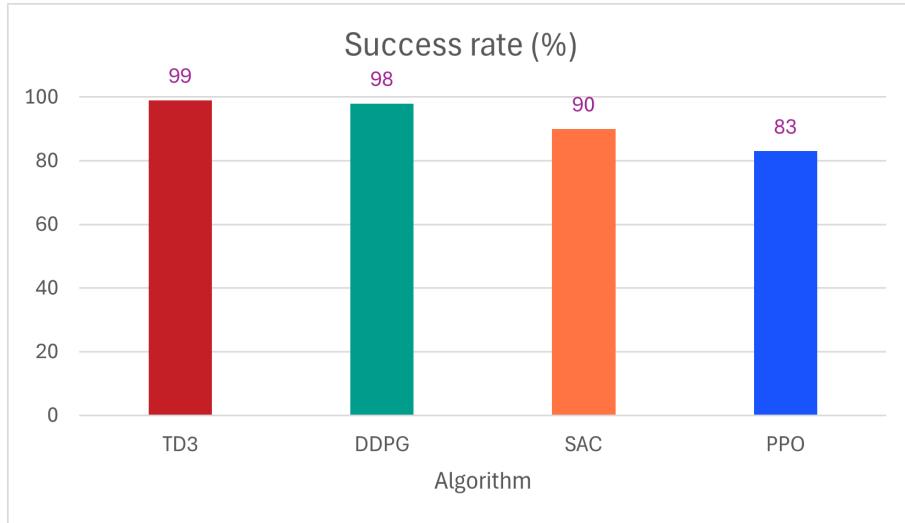
Οι συντελεστές στην παραπάνω εξίσωση αντιπροσωπεύουν τη βαρύτητα (%), που έχει η κάθε μετρική στο τελικό σκορ και επιλέχθηκαν αυθαίρετα. Έτσι, παρατηρούμε πως κάθε παίκτης μπορεί να κερδίσει έως 70 πόντους, με βάση το ποσοστό επιτυχίας του. Έπειτα, 20 πόντοι καθορίζονται από το μέσο αριθμό συγκρούσεων του παίκτη, στα επεισόδια όπου πετυχαίνει το στόχο του, χάνοντας 5 από αυτούς τους πόντους, κάθε φορά που ο μέσος αριθμός συγκρούσεων του αυξάνεται κατά μία μονάδα. Τέλος, οι υπόλοιποι 10 πόντοι, καθορίζονται από το μέσο χρόνο ολοκλήρωσης των επεισοδίων του παίκτη, στα οποία πετυχαίνει τον στόχο του. Συγκεκριμένα, όταν ο μέσος χρόνος του παίκτη, είναι μικρότερος ή ίσος των 10 δευτερολέπτων, τότε ο παίκτης θεωρείται γρήγορος και κερδίζει όλους τους πόντους. Αντίθετα, όταν ο μέσος χρόνος του παίκτη, είναι μεγαλύτερος των 20 δευτερολέπτων, τότε ο παίκτης θεωρείται αργός και δεν κερδίζει κανέναν από αυτούς τους πόντους. Στην περίπτωση που ο μέσος χρόνος του παίκτη, βρίσκεται ανάμεσα στα 10 και 20 δευτερολέπτα, τότε ο παίκτης κερδίζει τον αντίστοιχο αριθμό πόντων, στο διάστημα (0, 10). Με τον τρόπο αυτό, η τιμή 100 του σκορ αντιπροσωπεύει τον τέλειο παίκτη, που παρκάρει σε όλες τις περιπτώσεις, χωρίς να προκαλέσει καμία σύγκρουση και σε αρκετά συντόμο χρόνο.

Στις επόμενες υποενότητες, θα παρουσιάσουμε τις μετρικές αυτές, καθώς και το τελικό σκορ των αλγορίθμων, στα δύο τελευταία επίπεδα δυσκολίας του παιχνιδιού και θα σχολιάσουμε τα αποτελέσματα.

### 6.2.1 Επίπεδο δυσκολίας 3 - Άμεση στάθμευση

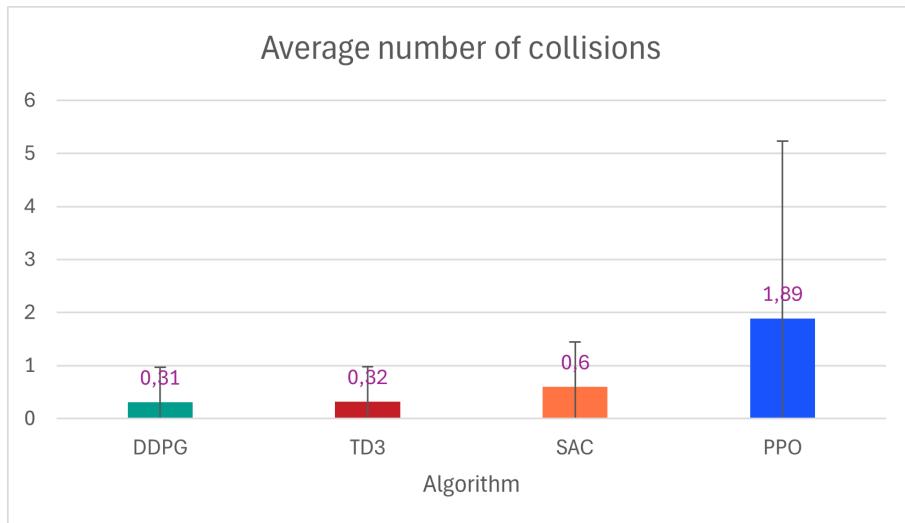
Στην Εικόνα 6.3, παρουσιάζονται τα ποσοστά επιτυχίας των πρακτόρων κάθε αλγορίθμου, στο επίπεδο δυσκολίας 3.

Παρατηρούμε πως όλοι οι αλγόριθμοι βαθιάς ενισχυτικής μάθησης πέτυχαν ποσοστό επιτυχίας άνω του 80%, ενώ οι αλγόριθμοι DDPG και TD3 ξεχωρίζουν, καθώς τα ποσοστά επιτυχίας τους προσεγγίζουν το 100%.



**Εικόνα 6.3.** Ποσοστά επιτυχίας των πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3.

Στην *Εικόνα 6.4*, παρουσιάζονται οι μέσοι αριθμοί συγκρούσεων των πρακτόρων κάθε αλγορίθμου, στο επίπεδο δυσκολίας 3.

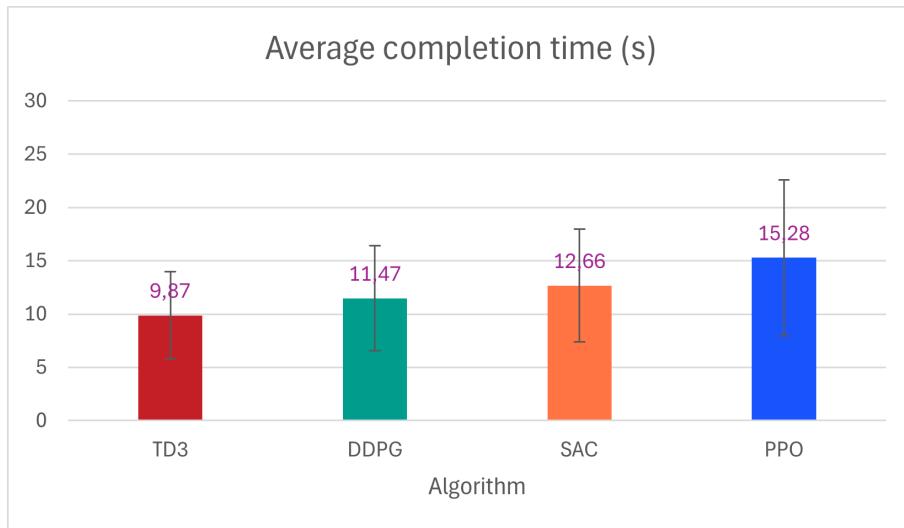


**Εικόνα 6.4.** Μέσοι αριθμοί συγκρούσεων των πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3.

Στο γράφημα αυτό, εκτός από τη μέση τιμή, απεικονίζεται και η τυπική απόκλιση ( $\sigma$ ) των μέσων αριθμών συγκρούσεων των πρακτόρων κάθε αλγορίθμου. Παρατηρούμε πως ξανά, οι αλγόριθμοι δράστη-κριτή έχουν τις καλύτερες επιδόσεις, με τους αλγορίθμους DDPG και TD3 να ξεχωρίζουν. Αντίθετα, ο αλγόριθμος PPO έχει υψηλό μέσο αριθμό συγκρούσεων, καθώς και τυπική απόκλιση αυτού. Επομένως, καταλαβαίνουμε πως η πολιτική που έχει αναπτύξει δεν είναι η βέλτιστη, αφού ακόμα και στα επεισόδια που πετυχαίνει τον στόχο του, προκαλεί πρώτα περίπου 2 συγκρούσεις.

## 6.2 Σύγκριση αλγορίθμων ως προς την απόδοση

Τέλος, στην *Εικόνα 6.5*, παρουσιάζονται οι μέσοι χρόνοι ολοκλήρωσης των πρακτόρων κάθε αλγορίθμου, στο επίπεδο δυσκολίας 3.

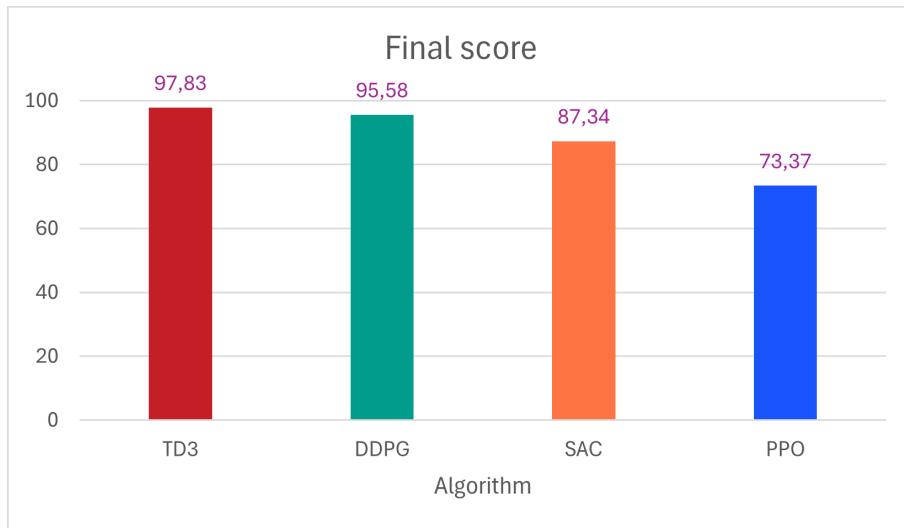


**Εικόνα 6.5.** Μέσοι χρόνοι ολοκλήρωσης των πρακτόρων κάθε αλγορίθμου στο επίπεδο δυσκολίας 3.

Παρατηρούμε πως οι μέσοι χρόνοι ολοκλήρωσης των πρακτόρων έχουν αρκετά χαμηλές τιμές, το οποίο αποτελεί θετική ένδειξη για την απόδοση τους. Ξανά, πρωταγωνιστεί ο αλγόριθμος TD3, ο οποίος χρειάζεται λιγότερο από το 1/3 του χρόνου του επεισοδίου για να το κερδίσει, ενώ ακόμα και ο αλγόριθμος με τη χειρότερη επίδοση (PPO), καταφέρνει να ολοκληρώσει το στόχο του σε μόλις 1/2 του χρόνου του επεισοδίου.

Έτσι, προκύπτουν τα τελικά σκορ των αλγορίθμων, για το επίπεδο δυσκολίας 3, τα οποία παρουσιάζονται στην *Εικόνα 6.6*.

Παρατηρούμε πως οι αλγόριθμοι TD3 και DDPG πετυχαίνουν τις καλύτερες επιδόσεις στο επίπεδο αυτό, φτάνοντας μάλιστα πολύ κοντά στο τέλειο σκορ. Ακολουθεί ο αλγόριθμος SAC με διαφορά 10 μονάδων, ενώ ο αλγόριθμος PPO καταλαμβάνει την τελευταία θέση, με σημαντική διαφορά από τους υπόλοιπους αλγορίθμους και εμφανή περιθώρια για βελτίωση στην πολιτική του. Επομένως, επιβεβαιώνεται από αυτό το γράφημα, η υπεροχή των αλγορίθμων δράστη-κριτή έναντι των απλούστερων αλγορίθμων βελτιστοποίησης πολιτικής, στην ικανότητα επίλυσης πολύπλοκων προβλημάτων.



**Εικόνα 6.6.** Τελικά σκορ των αλγορίθμων στο επίπεδο δυσκολίας 3.

### 6.2.2 Επίπεδο δυσκολίας 4 - Κανονική στάθμευση

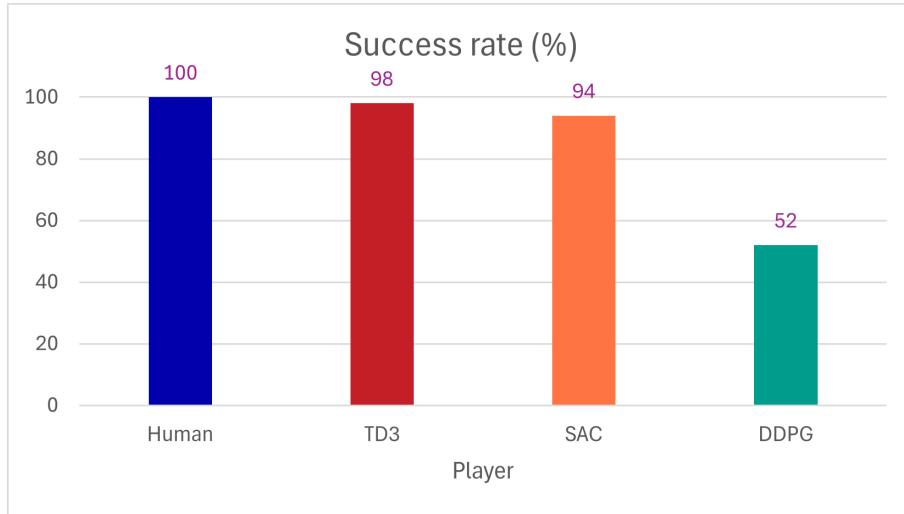
Στο επίπεδο αυτό, ελέγχεται η πραγματική ικανότητα των πρακτόρων να πάρκαρουν. Προκειμένου να έχουμε ένα πιο οικείο μέτρο σύγκρισης, της ικανότητας αυτής, επιλέξαμε να συγκρίνουμε την τελική απόδοση των πρακτόρων, ενάντια σε αυτήν ενός ανθρώπου, ειδικού στο παιχνίδι. Προσωπικά, μέσα από την κατασκευή του παιχνιδιού, καθώς και τις πολλές τροποποιήσεις που έγιναν κατά τη διαδικασία των εκπαιδεύσεων, έχω αποκτήσει μεγάλη εμπειρία στο παιχνίδι, κι έτσι οι επιδόσεις μου μπορούν να θεωρηθούν ως αυτές ενός ικανού ανθρώπινου παίκτη. Για αυτό, κατέγραψα τις επιδόσεις μου στα 100 επεισόδια αξιολόγησης και τις χρησιμοποίησα ως βάση σύγκρισης για τους πράκτορες.

Χρησιμοποιήθηκε ξανά, η εξίσωση 6.1 για τον υπολογισμό του τελικού σκορ, με τη διαφορά πως πλέον, ο χρόνος των 10 δευτερολέπτων, στον οποίο θεωρείται γρήγορη η ολοκλήρωση του επεισοδίου, έχει αυξηθεί στα 12, ώστε να συνυπολογίσουμε τα 2 επιπλέον δευτερόλεπτα της στάθμευσης. Με βάση αυτά, το τελικό ανθρώπινο σκορ υπολογίστηκε ίσο με 98.442. Αξίζει μάλιστα να σημειωθεί, πως μετά τον υπολογισμό των σκορ των πρακτόρων, αυτά κανονικοποιήθηκαν με βάση το ανθρώπινο σκορ, έτσι ώστε η τιμή 100 να μη συμβολίζει πλέον τη θεωρητικά, τέλεια επίδοση, αλλά την ανθρώπινη επίδοση.

Επομένως, στην Εικόνα 6.7, παρουσιάζονται τα ποσοστά επιτυχίας των παικτών στο επίπεδο δυσκολίας 4.

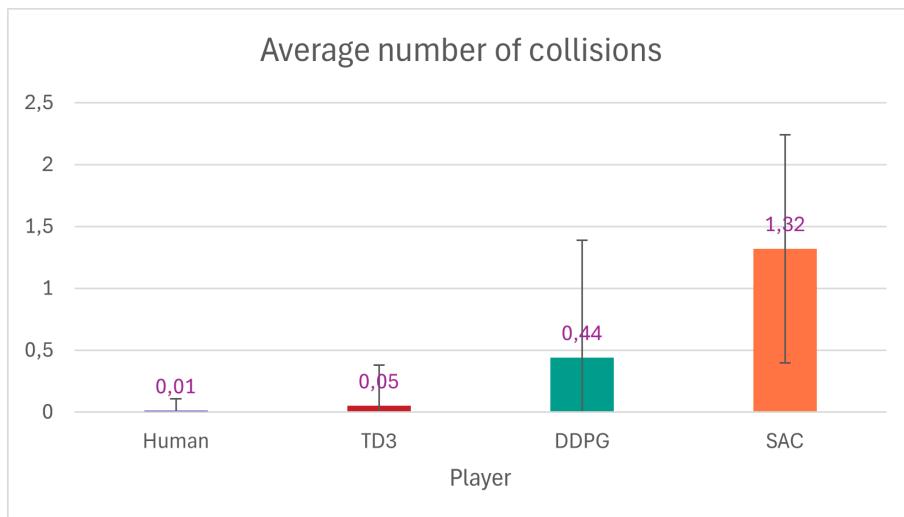
Παρατηρούμε πως το ανθρώπινο ποσοστό επιτυχίας ήταν ακριβώς 100%, ακολουθόμενο στενά από τους αλγορίθμους TD3 και SAC. Αντίθετα, το ποσοστό επιτυχίας του αλγορίθμου DDPG ήταν αρκετά χαμηλό, υποδεικνύοντας πως ο αλγόριθμος αυτός, δεν κατάφερε να λύσει ικανοποιητικά το πολύπλοκο πρόβλημα του επιπέδου δυσκολίας 4.

## 6.2 Σύγκριση αλγορίθμων ως προς την απόδοση



Εικόνα 6.7. Ποσοστά επιτυχίας των παικτών στο επίπεδο δυσκολίας 4.

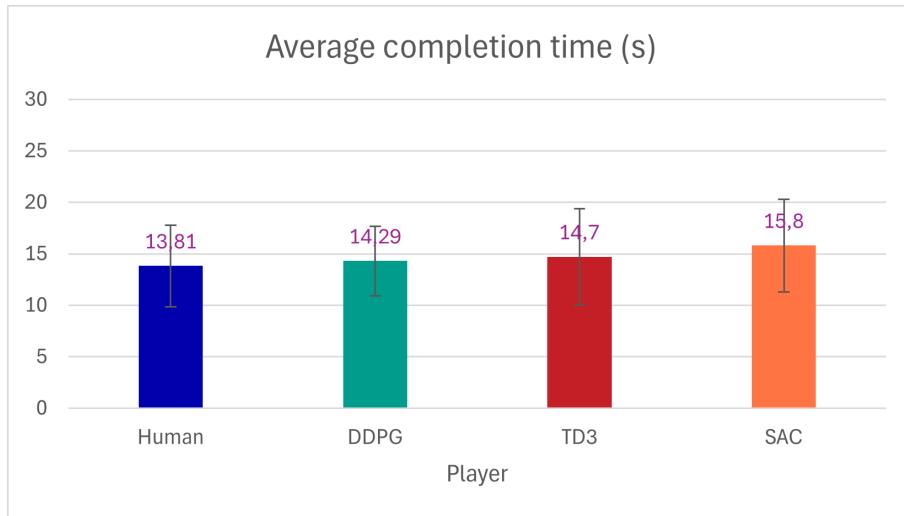
Στην Εικόνα 6.8, φαίνονται οι μέσοι αριθμοί συγκρούσεων των παικτών στο επίπεδο δυσκολίας 4.



Εικόνα 6.8. Μέσοι αριθμοί συγκρούσεων των παικτών στο επίπεδο δυσκολίας 4.

Από το συγκεκριμένο γράφημα επιβεβαιώνεται, πως ούτε οι άνθρωποι είναι τέλειοι, αφού σε ένα από τα 100 επεισόδια αξιολόγησης, προέκυψε μία σύγκρουση. Ο μέσος αριθμός συγκρούσεων του TD3 είναι συγκρίσιμος με τον ανθρώπινο, ενώ αυτός του DDPG είναι επίσης, αρκετά χαμηλός. Αυτό προκαλεί εντύπωση, καθώς βλέπουμε πως στα μισά επεισόδια, στα οποία ο αλγόριθμος καταφέρνει να παρκάρει, το κάνει με αποδοτικό τρόπο. Αντίθετα, η πολιτική που έχει αναπτύξει ο αλγόριθμος SAC είναι λιγότερο αποδοτική, καθώς ο πράκτορας του προκαλεί τουλάχιστον μία σύγκρουση, προτού παρκάρει.

Στην *Eικόνα 6.9*, φαίνονται οι μέσοι χρόνοι στάθμευσης των παικτών στο επίπεδο δυσκολίας 4.



**Εικόνα 6.9.** Μέσοι χρόνοι στάθμευσης των παικτών στο επίπεδο δυσκολίας 4.

Παρατηρούμε πως οι μέσοι χρόνοι στάθμευσης έχουν μικρή απόκλιση μεταξύ των παικτών, καθώς μόλις 2 δευτερόλεπτα χωρίζουν τον πρώτο (άνθρωπος) με τον τελευταίο (αλγόριθμος SAC). Ξανά, είναι αξιοσημείωτο το γεγονός πως ο DDPG, παρά το μικρό ποσοστό επιτυχίας του, πετυχαίνει τα καλύτερα αποτελέσματα μεταξύ των αλγορίθμων.

Έτσι, προκύπτουν τα τελικά, κανονικοποιημένα σκορ των αλγορίθμων, για το επίπεδο δυσκολίας 4, τα οποία παρουσιάζονται στην *Eικόνα 6.10*.

Παρατηρούμε πως το τελικό σκορ του αλγορίθμου TD3 (97.62), είναι πολύ κοντά στο ανθρώπινο σκορ. Είναι ενδιαφέρον, το πως ένας αλγόριθμος ενισχυτικής μάθησης, με μόλις 2.5 ημέρες εκπαίδευσης του (και μάλιστα, σε ένα αρκετά παρωχημένο υπολογιστικά, σύστημα), κατάφερε να προσεγγίσει τις ανθρώπινες επιδόσεις σε ένα τόσο πολύπλοκο πρόβλημα, όπως αυτό της στάθμευσης. Επίσης, αρκετά υψηλό είναι το σκορ του αλγορίθμου SAC, ενώ ακολουθεί με σημαντική διαφορά, ο αλγόριθμος DDPG, του οποίου η απλή προσέγγιση δεν στάθηκε ικανή, για την αξιόπιστη επίλυση του δυσκολότερου προβλήματος.



**Εικόνα 6.10.** Τελικά σκορ των αλγορίθμων στο επίπεδο δυσκολίας 4, κανονικοποιημένα ως προς τις ανθρώπινες επιδόσεις.

### 6.3 Συμπεράσματα

Τα τελικά συμπεράσματα της εργασίας, τα οποία προκύπτουν από τα αποτελέσματα της σύγκρισης των αλγορίθμων, συνοψίζονται στα ακόλουθα σημεία:

- Ο αλγόριθμος Q-Learning είναι ο πιο υπολογιστικά φτηνός αλγόριθμος που εξετάστηκε, καθώς συγκλίνει σε σημαντικά λιγότερο χρόνο, σε σχέση με τους αλγορίθμους βαθιάς ενισχυτικής μάθησης. Ωστόσο, επιβεβαιώνεται πως δεν είναι κατάλληλος για προβλήματα μεγάλης διαστασιμότητας, καθώς παρά τη διακριτοποίηση του περιβάλλοντος, δεν κατάφερε να εκπαιδευτεί αποτελεσματικά, στο πρόβλημα της αυτόματης στάθμευσης.
- Ο αλγόριθμος βελτιστοποίησης πολιτικής PPO είναι πιο αποδοτικός στον χρόνο εκπαίδευσης, σε σχέση με τους αλγορίθμους της κατηγορίας δράστη-κριτή, αφού η μέση διάρκεια εκπαίδευσης του (και άρα, ο χρόνος σύγκλισης του) είναι αρκετά μικρότερη. Ωστόσο, το πρόβλημα της αυτόματης στάθμευσης, αποδείχθηκε πολύ δύσκολο για τον αλγόριθμο, κι επομένως απαιτήθηκαν μεγάλο πλήθος προσπαθειών και σημαντικός χρόνος εκπαίδευσης, για να φτάσει σε επιθυμητά αποτελέσματα. Ακόμα όμως κι έτσι, οι επιδόσεις του ήταν χειρότερες σε σχέση με τους αλγορίθμους δράστη-κριτή και τα αποτελέσματα του στο πραγματικό πρόβλημα της αυτόματης στάθμευσης (επίπεδο 4), δεν ήταν ικανοποιητικά.
- Ο αλγόριθμος DDPG επαληθεύεται πως είναι ο πιο απλός, της κατηγορίας δράστη-κριτή, αφού χρειάστηκε το μικρότερο χρόνο εκπαίδευσης, σε σχέση με τους άλλους δύο, ενώ ο χρόνος αυτός ήταν συγκρίσιμος με τον αντίστοιχο του PPO. Παράλληλα, προσφέρει τις αυξημένες επιδόσεις των αλγορίθμων δράστη-κριτή, ειδικότερα σε λιγότερο σύνθετα περιβάλλοντα. Συγκεκριμένα,

στο πρόβλημα της άμεσης στάθμευσης, τα αποτελέσματα του DDPG ήταν εφάμιλλα με αυτά του καλύτερου αλγορίθμου (TD3), αλλά στο πρόβλημα της κανονικής στάθμευσης, οι επιδόσεις του αλγορίθμου μειώθηκαν δραστικά. Επομένως, ο αλγόριθμος DDPG κρίνεται ως μία καλή μέση λύση, μεταξύ χρόνου εκπαίδευσης και επιδόσεων και συνιστάται για απλούστερα προβλήματα.

- Ο αλγόριθμος SAC αποδείχθηκε πως βρίσκεται στο ενδιάμεσο μεταξύ του DDPG και του TD3, όσον αφορά τον απαιτούμενο χρόνο εκπαίδευσης του. Οι επιδόσεις του στα προβλήματα της άμεσης και της κανονικής στάθμευσης μπορεί να μην ήταν στο επίπεδο του TD3, ήταν όμως αρκετά υψηλές και σταθερές και στα δύο προβλήματα. Επομένως, ο αλγόριθμος SAC είναι μία αξιόπιστη λύση, ικανή να εφαρμοστεί ακόμα και σε πιο πολύπλοκα περιβάλλοντα.
- Ο αλγόριθμος TD3 επιβεβαιώνεται πως αποτελεί με διαφορά, τον πιο κοστοβόρο αλγόριθμο της κατηγορίας δράστη-κριτή. Παρόλα αυτά, οι επιδόσεις του αλγορίθμου, αντισταθμίζουν το αυξημένο υπολογιστικό του κόστος. Συγκεκριμένα, ο αλγόριθμος TD3 πέτυχε τα καλύτερα αποτελέσματα και στα δύο περιβάλλοντα που εξετάστηκαν. Μάλιστα, στο δυσκολότερο πρόβλημα της κανονικής στάθμευσης, οι επιδόσεις του αλγορίθμου προσέγγισαν με μεγάλη ακρίβεια τις ανθρώπινες. Επομένως, ο αλγόριθμος TD3, παρά τις απαιτήσεις του σε υπολογιστικούς πόρους, αποτελεί την καλύτερη επιλογή για το πρόβλημα της αυτόματης στάθμευσης.



## 7 Μελλοντικές Βελτιώσεις

Ο στόχος που τέθηκε για την παρούσα εργασία επετεύχθη, καθώς διεξήχθη μία τεκμηριωμένη σύγκριση διαφορετικών αλγορίθμων ενισχυτικής μάθησης. Μάλιστα, οι πράκτορες που αναπτύχθηκαν στα πλαίσια της εργασίας, έφτασαν σε επίπεδο απόδοσης που προσεγγίζει το ανθρώπινο. Ωστόσο, υπάρχουν πολλοί τρόποι με τους οποίους η εργασία αυτή μπορεί να βελτιωθεί ή και να επεκταθεί. Στη συνέχεια, παρουσιάζονται μερικές ιδέες για μελλοντική έρευνα και εξέλιξη της εργασίας, οι οποίες δεν υλοποιήθηκαν λόγω χρονικών περιορισμών.

Αρχικά, υπάρχει χώρος για βελτίωση στο κομμάτι του περιβάλλοντος των εκπαιδεύσεων. Συγκεκριμένα, μπορεί να αξιοποιηθεί κάποια βιβλιοθήκη φυσικής, ώστε να προσομοιωθούν με μεγαλύτερη ακρίβεια, μηχανισμοί όπως η κίνηση του αυτοκινήτου και οι συγκρούσεις του. Άκομα, χρήσιμη θα ήταν η υλοποίηση ενός πιο σύνθετου και ακριβέστερου τύπου αισθητήρα, σε σχέση με τις ακτίνες ραντάρ (*raycasts*), όπως για παράδειγμα οι ακτίνες κουτιού (*boxcasts*) ή σφαίρας (*spherecasts*). Τέλος, μπορεί να σχεδιαστεί ένας εξ ολοκλήρου νέος χάρτης παιχνιδιού, στον οποίο θα εξετάζεται το σαφώς δυσκολότερο πρόβλημα της παράλληλης στάθμευσης.

Στη συνέχεια, στο κομμάτι των αλγορίθμων ενισχυτικής μάθησης, μπορούν να δοκιμαστούν νέες μέθοδοι και εργαλεία. Για παράδειγμα, η τεχνική των βοηθητικών σημάτων ανταμοιβής (*auxiliary reward signals*), όπως το εγγενές σήμα της «περιέργειας» (*intrinsic curiosity*), μπορεί να χρησιμοποιηθεί για την ενθάρρυνση της εξερεύνησης του πράκτορα. Αντίστοιχα, η μέθοδος της επανάληψης εμπειρίας (*experience replay*), μπορεί να αυξήσει τη σταθερότητα και την αποδοτικότητα της εκπαίδευσης. Τέλος, η χρήση ενός λογισμικού αυτόματης βελτιστοποίησης υπερπαραμέτρων μηχανικής μάθησης, όπως το [Optuna](#), ίσως βελτιώσει τα αποτελέσματα της εκπαίδευσης.

Πέραν όμως των αλγορίθμων που ήδη αναλύθηκαν σε αυτήν την εργασία, το πρόβλημα της αυτόματης στάθμευσης μπορεί να επεκταθεί και σε άλλες κατηγορίες αλγορίθμων. Δύο πολύ ενδιαφέρουσες τεχνικές μηχανικής μάθησης που θα μπορούσαν να εφαρμοστούν, είναι η μάθηση με μίμηση (*imitation learning*) και οι γενετικοί αλγόριθμοι (*genetic algorithms*).

Στη μάθηση με μίμηση, ο πράκτορας αναπτύσσει την πολιτική του, αντιγράφοντας τη συμπεριφορά ενός ανθρώπου ειδικού, μέσω παρατηρήσεων ζευγών κατάστασης-ενέργειας. Ωστόσο, μέσα από αυτήν τη διαδικασία, ο πράκτορας μπορεί μόνο να αναπαράγει την απόδοση του ειδικού και όχι να την υπερβεί. Έτσι, πολλές φορές η μάθηση με μίμηση χρησιμοποιείται σε συνδυασμό με την ενισχυτική μάθηση, ώστε να επιταχυνθεί η διαδικασία εκπαίδευσης στα πρώτα στάδια της, αλλά και

---

να βελτιωθεί η τελική απόδοση του πράκτορα.

Από την άλλη, οι γενετικοί αλγόριθμοι είναι μία κατηγορία αλγορίθμων βελτιστοποίησης, οι οποίοι εμπνέονται από την εξέλιξη των ειδών στη φύση. Συγκεκριμένα, δημιουργείται ένας πληθυσμός υποψηφίων λύσεων, ο οποίος εξελίσσεται μέσα από διαδικασίες όπως η επιλογή (*selection*), η διασταύρωση (*crossover*) και η μετάλλαξη (*mutation*). Αποτελούν μία ιδιαίτερα δημοφιλή επιλογή για την εκπαίδευση πρακτόρων σε παιχνίδια αγώνων ταχύτητας (*racing games*), άλλα συναντώνται σπανιότερα σε προβλήματα αυτόματης στάθμευσης. Αυτοί μάλλον οφείλεται στη δυσκολία του συγκεκριμένου προβλήματος, αφού πρόκειται για ένα περιβάλλον με αραιές ανταμοιβές. Οι γενετικοί αλγόριθμοι δεν είναι κατάλληλοι για τέτοια περιβάλλοντα, καθώς η ανατροφοδότηση στους πράκτορες τους, δίνεται μόνο στο τέλος κάθε γενιάς και όχι μετά από κάθε ενέργεια, όπως στην ενισχυτική μάθηση. Ο παράγοντας αυτός, σε συνδυασμό με το αυξημένο υπολογιστικό κόστος τους, οδήγησε στην προτίμηση άλλων μεθόδων στην παρούσα εργασία. Παρόλα αυτά, ο πειραματισμός με αυτούς τους αλγορίθμους θα μπορούσε να αποτελέσει μία ενδιαφέρουσα κατεύθυνση για μελλοντική έρευνα.

## 8 Βιβλιογραφία

- AI-Forge. 2022. ‘A.I. Learns To Drive –youtube.com’ . [https://www.youtube.com/watch?v=ZshliCIM9ZA&ab\\_channel=AIForge](https://www.youtube.com/watch?v=ZshliCIM9ZA&ab_channel=AIForge).
- Alvarez, Waldo. 2017. ‘Markov decision process - Wikipedia –en.wikipedia.org’ . [https://en.wikipedia.org/wiki/Markov\\_decision\\_process](https://en.wikipedia.org/wiki/Markov_decision_process).
- Amid, Fish. 2018. ‘Lessons Learned Reproducing a Deep Reinforcement Learning Paper –amid.fish’ . <http://amid.fish/reproducing-deep-rl>.
- Anzalone, Luca, Paola Barra, Silvio Barra, Aniello Castiglione, και Michele Nappi. 2022. ‘An End-to-End Curriculum Learning Approach for Autonomous Driving Scenarios’. *IEEE Transactions on Intelligent Transportation Systems* 23 (10): 19817–26. <https://doi.org/10.1109/TITS.2022.3160673>.
- Arzt, Samuel. 2019. ‘AI Learns to Park - Deep Reinforcement Learning –youtube.com’ . [https://www.youtube.com/watch?v=VMp6pq6\\_QjI&t=462s&ab\\_channel=SamuelArzt](https://www.youtube.com/watch?v=VMp6pq6_QjI&t=462s&ab_channel=SamuelArzt).
- Atomwise. 2018. ‘AtomNet® Technology has the Power to Impact Early Drug Discovery – blog.atomwise.com’ . <https://blog.atomwise.com/atomnet-technology-has-the-power-to-impact-early-drug-discovery>.
- Baeldung. 2023. ‘Epsilon-Greedy Q-learning –baeldung.com’ . <https://www.baeldung.com/cs/epsilon-greedy-q-learning>.
- Blanco, Sebastian. 2023. ‘Report: Tesla Autopilot Involved in 736 Crashes since 2019 –caranddriver.com’ . <https://www.caranddriver.com/news/a44185487/report-tesla-autopilot-crashes-since-2019/>; CarandDriver.com.
- Blomqvist, Anders, και Christian Andersson. 2022. ‘Exploring the parameter space of Qlearning for faster convergence using Snake’ .
- BostonDynamics. 2024. ‘Starting on the Right Foot with Reinforcement Learning | Boston Dynamics –bostondynamics.com’ . <https://bostondynamics.com/blog/starting-on-the-right-foot-with-reinforcement-learning/>.
- Bright Side. 2020. ‘Self-Driving Cars: 7 Pros and 7 Cons –youtube.com’ . [https://www.youtube.com/watch?v=9RAFgKcY4uA&ab\\_channel=BRIGHTSIDE](https://www.youtube.com/watch?v=9RAFgKcY4uA&ab_channel=BRIGHTSIDE).

- 
- Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, και Wojciech Zaremba. 2016. ‘OpenAI Gym’ . <https://arxiv.org/abs/1606.01540>.
- Code-Bullet. 2019. ‘A.I. Learns to DRIVE –youtube.com’ . [https://www.youtube.com/watch?v=r428O\\_CMcpI&ab\\_channel=CodeBullet](https://www.youtube.com/watch?v=r428O_CMcpI&ab_channel=CodeBullet).
- Comsol. 2023. ‘Deep Neural Network –doc.comsol.com’ . [https://doc.comsol.com/6.2/doc/com.comsol.help.comsol/comsol\\_ref\\_definitions.19.050.html](https://doc.comsol.com/6.2/doc/com.comsol.help.comsol/comsol_ref_definitions.19.050.html).
- Craft, Marcus. 2023. ‘Average Length of Car - How long is a Standard Motor Vehicle? – carsguide.com.au’ . <https://www.carsguide.com.au/car-advice/whats-the-average-length-of-a-car-89454>.
- DeepMind. 2016. ‘AlphaGo –deepmind.google’ . <https://deepmind.google/technologies/alphago/>.
- Deichmann, Johannes, Eike Ebel, Kersten Heineke, Ruth Heuss, Martin Kellner, και Fabian Steiner. 2023. ‘Autonomous driving’ s future: Convenient and connected –mckinsey.com’ . <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected>; McKinsey.
- Dey, Victor. 2023. ‘The debate over neural network complexity: Does bigger mean better? – venturebeat.com’ . <https://venturebeat.com/ai/neural-network-complexity-is-it-getting-better/>.
- Doss, Minnie. 2024. ‘Vehicle Turning Path Layouts Dimensions & Drawings | Dimensions.com – dimensions.com’ . <https://www.dimensions.com/collection/vehicle-turning-path-layouts>.
- EITCA. 2024. ‘Does increasing of the number of neurons in an artificial neural network layer increase the risk of memorization leading to overfitting? - EITCA Academy –eitca.org’ . <https://eitca.org/artificial-intelligence/eitc-ai-tff-tensorflow-fundamentals/overfitting-and-underfitting-problems/solving-models-overfitting-and-underfitting-problems-part-1/does-increasing-of-the-number-of-neurons-in-an-artificial-neural-network-layer-increase-the-risk-of-memorization-leading-to-overfitting/>.
- Erickson, Jim. 2018. ‘Maximizing the environmental benefits of autonomous vehicles –news.umich.edu’ . <https://news.umich.edu/maximizing-the-environmental-benefits-of-autonomous-vehicles/>; University of Michigan.
- Ezeokeke, Emmanuel. 2024. ‘AI Agents vs. AI Models: Why Agents Take the Lead –linkedin.com’ . <https://www.linkedin.com/pulse/ai-agents-vs-models-why-take-lead-emmanuel-ezeokeke-gv4xf>; LinkedIn.
- Fujimoto, Scott, Herke Hoof, και David Meger. 2018. ‘Addressing Function Approximation Error in Actor-Critic Methods’ . <https://arxiv.org/abs/1802.09477>.
- Gillis, Alexander. 2024. ‘What is a self-driving car? | Definition from TechTarget –techtarget.com’ . <https://www.techtarget.com/searchenterpriseai/definition/driverless-car>; TechTarget.

- 
- Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, και Sergey Levine. 2018. ‘Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor’ . <https://arxiv.org/abs/1801.01290>.
- Hanson Robotics. 2016. ‘Sophia - Hanson Robotics –hansonrobotics.com’ . <https://www.hansonrobotics.com/sophia/>.
- Irpan, Alex. 2018. ‘Deep Reinforcement Learning Doesn’t Work Yet’ . <https://www.alexirpan.com/2018/02/14/rl-hard.html>.
- Johnson, Jonathan. 2020. ‘What’s a Deep Neural Network? Deep Nets Explained –bmc.com’ . <https://www.bmc.com/blogs/deep-neural-network/>; BMC Software.
- Karpathy, Andrej. 2016. ‘Neural Networks Part 1: Setting Up the Architecture. Notes for CS231n Convolutional Neural Networks for Visual Recognition’ . Stanford University: Stanford, CA, USA.
- Lai, Leonardo. 2018. ‘Automatic Parking with Q-Learning’ . GitHub repository. MA. thesis, Sant’ Anna School of Advanced Studies, Pisa; <https://github.com/leoll2/Autoparking>. <https://doi.org/10.5281/zenodo.4568892>.
- Lateef, Zulaikha. 2024. ‘Types of AI: Understanding Different Types of Artificial Intelligence in 2024 –edureka.co’ . <https://www.edureka.co/blog/types-of-artificial-intelligence/>; Edureka.
- Lee, Dan. 2019. ‘Reinforcement Learning, Part 1: A Brief Introduction –medium.com’ . <https://medium.com/ai%C2%B3-theory-practice-business/reinforcement-learning-part-1-a-brief-introduction-a53a849771cf>.
- Leger, Corentin, και Antonin Raffin. 2024. ‘Reinforcement Learning Tips and Tricks &#x2014; Stable Baselines3 2.4.0a8 documentation –stable-baselines3.readthedocs.io’ . [https://stable-baselines3.readthedocs.io/en/master/guide/rl\\_tips.html](https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html).
- Lillicrap, Timothy, Jonathan Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, και Daan Wierstra. 2015. ‘Continuous control with deep reinforcement learning’ . <https://arxiv.org/abs/1509.02971>.
- Loiacono, Daniele, Alessandro Prete, Pier Luca Lanzi, και Luigi Cardamone. 2010. ‘Learning to overtake in TORCS using simple reinforcement learning’ . Στο IEEE Congress on Evolutionary Computation, 1–8. <https://doi.org/10.1109/CEC.2010.5586191>.
- McCarthy, John. 2004. ‘What is Artificial Intelligence?’
- McMurray, Alex. 2023. ‘Goldman Sachs AI head likes technique that creates a trader’ –efinancialcareers.com’ . <https://www.efinancialcareers.com/news/2023/10/will-ai-replace-traders-goldman-sachs>.
- Melo, Francisco. 2001. ‘Convergence of Q-learning: a simple proof’ . <http://users.isr.ist.utl.pt/~mtjspaan/readingGroup/ProofQlearning.pdf>.

- 
- Metz, Cade. 2016. ‘In Two Moves, AlphaGo and Lee Sedol Redefined the Future —wired.com’ . <https://www.wired.com/2016/03/two-moves-alphago-lee-sedol-redefined-future/>.
- Miles, Robert. 2024. ‘What is reward hacking? —ui.stampy.ai’ . <https://ui.stampy.ai/questions/8SIU/What-is-reward-hacking>; AI Safety.info.
- Mitchell, Thomas. 1997. *Machine Learning*. McGraw-Hill series in computer science. New York, NY: McGraw-Hill Professional.
- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, κατεξόδιο. 2015. ‘Human-level control through deep reinforcement learning’ . *Nature* 518: 529–33. <https://api.semanticscholar.org/CorpusID:205242740>.
- Moreira, Dinis. 2021. ‘Deep Reinforcement Learning for Automated Parking’ . MA. thesis, University of Porto; <https://repositorio-aberto.up.pt/handle/10216/136074>.
- Mujumdar, Pranav, Parthiv Shah, κατεξόδιο Xinyue Cui. 2020. ‘Autonomous Car Parking Simulator using Unity MLAgents’ . MA. thesis, Northeastern University; <https://github.com/pranavmujumdar/CarparkingAi>.
- NeuralNine. 2021. ‘Self-Driving AI Car Simulation in Python —youtube.com’ . [https://www.youtube.com/watch?v=Cy155O5R1Oo&ab\\_channel=NeuralNine](https://www.youtube.com/watch?v=Cy155O5R1Oo&ab_channel=NeuralNine).
- OpenAI. 2018. ‘Part 2: Kinds of RL Algorithms &#x2014; Spinning Up documentation — spinningup.openai.com’ . [https://spinningup.openai.com/en/latest/spinningup/rl\\_intro2.html](https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html).
- OpenAI. 2019. ‘OpenAI Five defeats Dota 2 world champions’ . <https://openai.com/index/openai-five-defeats-dota-2-world-champions/>.
- Papers With Code. 2024. ‘Papers with Code - PPO Explained —paperswithcode.com’ . <https://paperswithcode.com/method/ppo>.
- Parkinson, Avery. 2019. ‘The Epsilon-Greedy Algorithm for Reinforcement Learning —medium.com’ . <https://medium.com/analytics-vidhya/the-epsilon-greedy-algorithm-for-reinforcement-learning-5fe6f96dc870>; Analytics Vidhya.
- Patel, Purvag G., Norman Carver, κατεξόδιο Shahram Rahimi. 2011. ‘Tuning computer gaming agents using Q-learning’ . Στο 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), 581–88.
- Patterson, Andrew, Samuel Neumann, Martha White, κατεξόδιο Adam White. 2023. ‘Empirical Design in Reinforcement Learning’ . <https://arxiv.org/abs/2304.01315>.
- Porro, Marco. 2020. ‘Self Driving Car Neural Network - with Python and NEAT (CODE in the description) —youtube.com’ . [https://www.youtube.com/watch?v=cFjYinc465M&ab\\_channel=MarcoPorro](https://www.youtube.com/watch?v=cFjYinc465M&ab_channel=MarcoPorro).

- 
- Prijono, Benny. 2020. ‘GitHub - bennylp/RL-Taxonomy: Loose taxonomy of reinforcement learning algorithms –github.com’ . <https://github.com/bennylp/RL-Taxonomy>.
- Puigdomenech, Adria, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, και Charles Blundell. 2020. ‘Agent57: Outperforming the human Atari benchmark – deepmind.google’ . <https://deepmind.google/discover/blog/agent57-outperforming-the-human-atari-benchmark/>.
- Quang, Luu. 2024. ‘Q-Learning vs. Deep Q-Learning vs. Deep Q-Network | Baeldung on Computer Science –baeldung.com’ . [https://www.baeldung.com/cs/q-learning-vs-deep-q-learning-vs-deep-q-network?fbclid=IwZXh0bgNhZW0CMTEAAR2SFVuY7XwXuXgcuva0kKQ24Y\\_xpBt4A6Ajsq79KfHwmyvdLOsOQD1MrmM\\_aem\\_6t-0YIbksWbZFNA1XWkNhw](https://www.baeldung.com/cs/q-learning-vs-deep-q-learning-vs-deep-q-network?fbclid=IwZXh0bgNhZW0CMTEAAR2SFVuY7XwXuXgcuva0kKQ24Y_xpBt4A6Ajsq79KfHwmyvdLOsOQD1MrmM_aem_6t-0YIbksWbZFNA1XWkNhw).
- Raffin, Antonin. 2021. ‘- YouTube –youtube.com’ . [https://www.youtube.com/watch?v=Ikngt0\\_DXJg](https://www.youtube.com/watch?v=Ikngt0_DXJg).
- Russell, Stuart, και Peter Norvig. 2021. *Artificial intelligence: A modern approach, global edition*. 4ο έκδ. London, England: Pearson Education.
- Saini, Sajan. 2019. ‘How do self-driving cars 'see'? - Sajan Saini –youtube.com’ . [https://www.youtube.com/watch?v=PRg5RNU\\_JLk&ab\\_channel=TED-Ed](https://www.youtube.com/watch?v=PRg5RNU_JLk&ab_channel=TED-Ed).
- Samuel, A. L. 1959. ‘Some studies in machine learning using the game of checkers’ . *IBM Journal of Research and Development* 44 (1.2): 206–26. <https://doi.org/10.1147/rd.441.0206>.
- Saravia, Felipe. 2020. ‘Neural Network Cars and Genetic Algorithms (1/2) –youtube.com’ . [https://www.youtube.com/watch?v=-sg-GgoFCP0&ab\\_channel=FelipeSaravia](https://www.youtube.com/watch?v=-sg-GgoFCP0&ab_channel=FelipeSaravia).
- Schulman, John. 2017. ‘- YouTube –youtube.com’ . <https://www.youtube.com/watch?v=8EcdaCk9KaQ>.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, και Oleg Klimov. 2017. ‘Proximal Policy Optimization Algorithms’ . <https://arxiv.org/abs/1707.06347>.
- Schumacher, Devin. 2023. ‘Generalized State-Dependent Exploration’ . <https://serp.ai/generalized-state-dependent-exploration/>.
- Singh, S. 2015. ‘Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey’ . <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115>; NHTSA’s National Center for Statistics; Analysis.
- Society of Automobile Engineers. 2021. ‘SAE Levels of Driving Automation™ Refined for Clarity and International Audience –sae.org’ . <https://www.sae.org/blog/sae-j3016-update>.
- Stewart, Ellis. 2023. ‘What is Machine Learning (ML)? Types, Models, Algorithms | Enterprise Tech News EM360 –em360tech.com’ . <https://em360tech.com/tech-article/what-is-machine-learning-ml>; EM360 Tech.

- 
- Sutton, Richard S, και Andrew G Barto. 2018. *Reinforcement learning: An introduction*. 2ο έκδ. Cambridge, MA: MIT press.
- Szymanski, Lech. 2018. ‘Course COSC470’ . <https://www.cs.otago.ac.nz/cosc470/>; University of Otago, Computer Science Department.
- Tesauro, Gerald. 1994. ‘TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play’ . *Neural Computation* 6 (2): 215–19. <https://doi.org/10.1162/neco.1994.6.2.215>.
- Thomson, T., και Ryan Thomas. 2023. ‘Ageism, sexism, classism and more: 7 examples of bias in AI-generated images —theconversation.com’ . <https://theconversation.com/ageism-sexism-classism-and-more-7-examples-of-bias-in-ai-generated-images-208748#:~:text=There%20were%20also%20notable%20differences,of%20more%20fluid%20gender%20expression>.
- Trudell, Craig. 2024. ‘Bloomberg - Are you a robot? —bloomberg.com’ . <https://www.bloomberg.com/news/newsletters/2024-04-25/elon-musk-s-tesla-robotaxi-predictions-were-all-wrong>; Bloomberg.
- Tucker, Sean. 2024. ‘Self-Driving Cars: Everything You Need To Know - Kelley Blue Book —kbb.com’ . <https://www.kbb.com/car-advice/self-driving-cars/>; Kelley Blue Book.
- Turing, A. M. 1950. ‘I.—COMPUTING MACHINERY AND INTELLIGENCE’ . *Mind* LIX (236): 433–60. <https://doi.org/10.1093/mind/LIX.236.433>.
- University of Michigan Center for Sustainable Systems. 2023. ‘Autonomous Vehicles Factsheet —css.umich.edu’ . <https://css.umich.edu/publications/factsheets/mobility/autonomous-vehicles-factsheet>.
- Watkins, Christopher. 1989. ‘Learning From Delayed Rewards’ , Ιανουάριος.
- Wikipedia. 2024. ‘Multi-armed bandit - Wikipedia —en.wikipedia.org’ . [https://en.wikipedia.org/wiki/Multi-armed\\_bandit#cite\\_note-Whittle79-15](https://en.wikipedia.org/wiki/Multi-armed_bandit#cite_note-Whittle79-15).
- Yosh. 2022. ‘- YouTube —youtube.com’ . <https://www.youtube.com/watch?v=SX08NT55YhA&t=29s>.
- Zand, Aria, Zack Stokes, Arjun Sharma, Welmoed van Deen, και Daniel Hommes. 2022. ‘Artificial Intelligence for Inflammatory Bowel Diseases (IBD); Accurately Predicting Adverse Outcomes Using Machine Learning’ . *Digestive Diseases and Sciences* 67 (Απρίλιος): 1–12. <https://doi.org/10.1007/s10620-022-07506-8>.