ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ – ΜΗΧΑΝΙΚΩΝ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ «ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σύγκριση Μεθόδων Deep Learning και Reinforcement Learning για την επίλυση του προβλήματος του Περιοδεύοντος Πωλητή.

Γεώργιος Κ. Μπούγας
Α.Μ. 60202315

Εισηγητής: Βασίλειος Βεσκούκης

(Υπογραφή)

……

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σύγκριση Μεθόδων Deep Learning και Reinforcement Learning για την επίλυση του προβλήματος του Περιοδεύοντος Πωλητή

Γεώργιος Κ. Μπούγας
Α.Μ. 60202315

Εισηγητής:

Βασίλειος Βεσκούκης

Εξεταστική Επιτροπή:

Αναστάσιος Δουλάμης
Νικόλαος Δουλάμης

Ημερομηνία εξέτασης:

18/02/2022

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω αρχικά τον διευθυντή κ. Μαρίνο Κάβουρα που μου έδωσε την ευκαιρία να παρακολουθήσω το συγκεκριμένο πρόγραμμα μεταπτυχιακών σπουδών για το οποίο είχα ακούσει μόνο καλά λόγια που επαληθεύτηκαν. Τον καθηγητή μου κ. Βασίλειο Βεσκούκη που με επέβλεψε, με υποστήριξε αλλά και με άφησε ελεύθερο κατά την συγγραφή της παρούσας μεταπτυχιακής εργασίας, όπως επίσης και τους υπόλοιπους καθηγητές των οποίων τα μαθήματα είχα την τιμή να παρακολουθήσω που μέσα από πρωτόγνωρες συνθήκες εξαιτίας της πανδημίας, κατάφεραν και μετέδωσαν τις πολύτιμες γνώσεις τους με μεγάλη χαρά και ευγένεια.

Τέλος θα ήθελα να ευχαριστήσω την μέντορα μου και καλύτερή μου φίλη που χωρίς αυτήν δεν θα είχα προσπαθήσει για την εισαγωγή μου στο πρόγραμμα, για την αμέριστη, ανιδιοτελή στήριξη και ώθηση που μου έδωσε καθ' όλη την διάρκεια των σπουδών μου, που είναι πάντα δίπλα μου, πίστεψε σε μένα και με έκανε να πιστέψω και εγώ στον εαυτό μου.

Μην πεις ποτέ σου «είναι αργά»
-Always
Γιώργος

ΠΕΡΙΛΗΨΗ

Στην καθημερινότητα μας καλούμαστε να επιλέξουμε τις συντομότερες διαδρομές για ποικίλες δραστηριότητες που ερχόμαστε αντιμέτωποι. Είναι στη φύση μας να θέλουμε να ολοκληρώσουμε οποιαδήποτε διαδικασία όσο το δυνατόν γρηγορότερα και με το λιγότερο κουραστικό τρόπο. Όλα αυτά που περιγράψαμε συναντώνται στο πρόβλημα του περιοδεύοντος πωλητή (TSP) το οποίο αποτελεί το πιο κοινό πρόβλημα συνδυαστικής βελτιστοποίησης που καλούμαστε διερευνήσουμε. Η παρούσα μεταπτυχιακή εργασία έχει ως σκοπό να ερευνήσει τις πρόσφατες προσπάθειες, τόσο από τις κοινότητες μηχανικής μάθησης όσο και από τις κοινότητες επιχειρησιακής έρευνας, για την εκμετάλλευση μεθόδων μηχανικής μάθησης στην επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης. Η συνδυαστική βελτιστοποίηση (CO) είναι το εργαλείο πολλών σημαντικών εφαρμογών στην επιχειρησιακή έρευνα, τη μηχανική και άλλους τομείς και, ως εκ τούτου, έχει προσελκύσει τεράστια προσοχή από την ερευνητική κοινότητα πρόσφατα. Δεδομένης της δύσκαμπτης φύσης αυτών των προβλημάτων, οι αλγόριθμοι τελευταίας τεχνολογίας βασίζονται σε χειροποίητες ευρετικές για τη λήψη αποφάσεων που κατά τα άλλα είναι πολύ δαπανηρές για να υπολογιστούν ή δεν καθορίζονται ορθά από μαθηματική άποψη. Έτσι, η μηχανική μάθηση φαίνεται σαν ένας φυσικός υποψήφιος για τη λήψη τέτοιων αποφάσεων με έναν πιο στιβαρό και βελτιστοποιημένο τρόπο. Γενικά, θεωρούμε αρκετά σημαντική την περαιτέρω προώθηση της ενσωμάτωσης της μηχανικής μάθησης και της συνδυαστικής βελτιστοποίησης και θα περιγράψουμε λεπτομερώς μια μεθοδολογία για να στηρίξουμε αυτή τη θέση.

Όπως αναφέρθηκε παραπάνω, ορισμένες αποτελεσματικές προσεγγίσεις σε κοινά προβλήματα περιλαμβάνουν τη χρήση χειροποίητων ευρετικών για τη διαδοχική κατασκευή μιας λύσης. Ως εκ τούτου, είναι ενδιαφέρον να δούμε πώς ένα πρόβλημα CO και πιο συγκεκριμένα το πρόβλημα TSP μπορεί να αναδιατυπωθεί ως μια διαδοχική διαδικασία λήψης αποφάσεων και εάν αυτές οι ευρετικές μπορούν να μαθευτούν στο παρασκήνιο από έναν πράκτορα ενισχυτικής μάθησης (RL).

Εν κατακλείδι, η παρούσα εργασία θα διερευνήσει και τη συνέργεια μεταξύ των πλαισίων CO και RL, η οποία μπορεί να γίνει μια πολλά υποσχόμενη κατεύθυνση για την επίλυση συνδυαστικών προβλημάτων.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Βαθιά Μάθηση, Ενισχυμένη Μάθηση, Συνδυαστική Βελτιστοποίηση, Πρόβλημα του Περιοδεύοντος Πωλητή.

ABSTRACT

We are called to choose the shortest routes for various activities in our daily lives. It is in our nature to want to complete any process as quickly as possible and in the least tedious way. Everything we have described is encountered in the traveling salesman (TSP) problem, the most common combinatorial optimization problem. The present dissertation aims to investigate recent efforts, both by the machine learning communities and by the operational research communities, to exploit machine learning methods in solving combinatorial optimization problems. Combinatorial optimization (CO) is the tool of many important applications in operational research, engineering, and other fields and, therefore, has attracted a lot of attention from the research community recently. Given the rigid nature of these problems, state-of-the-art algorithms rely on handcrafted decision-making heuristics that are otherwise too expensive to calculate or determine mathematically. Thus, machine learning seems like a natural candidate for making such decisions in a more robust and optimized way. In general, we consider it quite important to further promote the integration of machine learning and combinatorial optimization and we will describe in detail a methodology to support this position. As mentioned above, some effective approaches to common problems include the use of handcrafted heuristics to sequentially construct a solution. Therefore, it is interesting to see how a CO problem, especially the TSP problem, can be restructured as a sequential decision-making process and whether these heuristics can be learned in the background by an auxiliary learning agent (RL). In conclusion, our work will also explore the synergy between the CO and RL frameworks, which can become a very promising direction for solving combinatorial optimization problems.

KEYWORDS: Machine Learning, Deep Reinforcement Learning, Combinatorial Optimization, Traveling Salesman problem.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Οι ορισμοί που ακολουθούν με τις μεταφράσεις και τα ακρωνύμια τους, είναι προσαρμοσμένοι προκειμένου να διευκολύνουν την ανάγνωση του συγκεκριμένου τόμου και προέρχονται από διάφορα λεξιλόγια και πηγές, στα οποία γίνεται αναφορά όπου αυτό κρίνεται απαραίτητο.

CO Combinatorial Optimization **-** Συνδυαστική Βελτιστοποίηση

ML Machine Learning – Μηχανική Μάθηση

RL Reinforcement Learning – Ενισχυμένη Μάθηση

DL Deep Learning – Βαθιά Μάθηση

MTA Meta Reinforcement Learning

TSP Travelling Salesman Problem – Πρόβλημα Περιοδεύοντος Πωλητή

RNN Recurrent Neural Network – Αναδρομικό Νευρωνικό Δίκτυο

CNN Convolutional Neural Network - Συνελικτικό Νευρωνικό Δίκτυο

FNN Feedback Neural Network – Νευρωνικά δίκτυα τροφοδοσίας

ANN Artificial Neural Network - Τεχνητά νευρωνικά δίκτυα

LSTM Long short term Memory - Μακροπρόθεσμη-Βραχυπρόθεσμη Μνήμη

GPN Graph Pointer Network

MDP Markov decision process – Διαδικασία Αποφάσεων Markov

AI Artificial Intelligence – Τεχνητή Νοημοσύνη

ΚΕΦΑΛΑΙΟ 1

1. Εισαγωγή

Η επιχειρησιακή έρευνα (operational research) ξεκίνησε κατά τον δεύτερο παγκόσμιο πόλεμο ως μια πρωτοβουλία για τη χρήση των μαθηματικών και της επιστήμης των υπολογιστών ώστε να βοηθήσουν τους στρατιωτικούς αναλυτές στις αποφάσεις τους (Fortun & Schweber, 1993). Σήμερα, χρησιμοποιείται ευρέως στη βιομηχανία, συμπεριλαμβανομένων, ενδεικτικά, των μεταφορών, της αλυσίδας εφοδιασμού, της ενέργειας, της χρηματοδότησης και του προγραμματισμού. Σε αυτή την εργασία, εστιάζουμε σε διακριτά προβλήματα βελτιστοποίησης που διατυπώνονται ως βελτιστοποίηση περιορισμένης ακεραιότητας, δηλαδή με ακέραιες ή δυαδικές μεταβλητές που ονομάζονται μεταβλητές απόφασης. Αν και δεν είναι δύσκολο να λυθούν όλα αυτά τα προβλήματα (π.χ. προβλήματα συντομότερης διαδρομής), επικεντρωνόμαστε σε προβλήματα συνδυαστικής βελτιστοποίησης (NP-hard). Αυτό πρακτικά δεν είναι και τόσο εύκολο, με την έννοια ότι, για αυτά τα προβλήματα, θεωρείται απίθανο να υπάρχει ένας αλγόριθμος του οποίου ο χρόνος εκτέλεσης είναι πολυωνυμικός ως προς το μέγεθος της εισόδου. Ωστόσο, στην πράξη, οι αλγόριθμοι συνδυαστικής βελτιστοποίησης μπορούν να λύσουν περιπτώσεις με έως και εκατομμύρια μεταβλητές απόφασης και περιορισμούς.

Πώς είναι δυνατόν να λυθούν προβλήματα NP-hard σε πρακτικό χρόνο; Ας δούμε το παράδειγμα του προβλήματος του περιοδεύοντος πωλητή, ένα πρόβλημα NP-hard που ορίζεται σε ένα γράφημα όπου αναζητούμε έναν κύκλο ελάχιστης διάρκειας όπου ο πωλητής επισκέπτεται μία και μόνο φορά κάθε κόμβο (δηλαδή πόλη). Μια ιδιαίτερη περίπτωση είναι αυτή του προβλήματος του Ευκλείδειου περιοδεύοντος πωλητή. Σε αυτή την έκδοση, σε κάθε κόμβο εκχωρούνται συντεταγμένες σε ένα επίπεδο και το κόστος σε μια άκρη που συνδέει δύο κόμβους είναι η Ευκλείδεια απόσταση μεταξύ τους. Αν και θεωρητικά είναι τόσο δύσκολο όσο το γενικό πρόβλημα του περιοδεύοντος πωλητή, η καλή κατά προσέγγιση λύση μπορεί να βρεθεί πιο αποτελεσματικά στην Ευκλείδεια περίπτωση αξιοποιώντας τη δομή του γραφήματος (Larson and Odoni, 1981).

Ομοίως, διάφορα είδη προβλημάτων επιλύονται με τη μόχλευση της ειδικής δομής τους. Άλλοι αλγόριθμοι, σχεδιασμένοι να είναι γενικοί, φαίνεται εκ των υστέρων ότι είναι εμπειρικά πιο αποτελεσματικοί σε συγκεκριμένους τύπους προβλημάτων. Η επιστημονική βιβλιογραφία καλύπτει το πλούσιο σύνολο τεχνικών που έχουν αναπτύξει οι ερευνητές για την αντιμετώπιση διαφορετικών προβλημάτων συνδυαστικής βελτιστοποίησης. Ένας ειδικός θα ξέρει πώς να βελτιώσει περαιτέρω τις παραμέτρους του αλγορίθμου σε διαφορετικές συμπεριφορές της διαδικασίας βελτιστοποίησης, επεκτείνοντας έτσι αυτή τη γνώση. Αυτές οι τεχνικές, και οι παράμετροι που τις ελέγχουν, έχουν μάθει συλλογικά να αποδίδουν στην απρόσιτη κατανομή περιπτώσεων προβλημάτων. Η εστίαση αυτής της εργασίας εντοπίζεται στους αλγόριθμους συνδυαστικής βελτιστοποίησης που εκτελούν αυτόματα τη μάθηση σε μια επιλεγμένη άρρητη κατανομή προβλημάτων. Η ενσωμάτωση στοιχείων μηχανικής μάθησης στον αλγόριθμο μπορεί να το πετύχει αυτό αποτελεσματικά.

Η μηχανική εκμάθηση εστιάζει στην εκτέλεση μιας εργασίας με (πεπερασμένα και συνήθως «θορυβώδη») δεδομένα. Είναι κατάλληλη για φυσικά σήματα για τα οποία δεν προκύπτει σαφής μαθηματική διατύπωση, επειδή η πραγματική κατανομή δεδομένων δεν είναι γνωστή, όπως κατά την επεξεργασία εικόνων, κειμένου, φωνής ή με συστήματα συστάσεων, κοινωνικά δίκτυα ή οικονομικές προβλέψεις. Τις περισσότερες φορές, το μαθησιακό πρόβλημα έχει μια στατιστική διατύπωση που επιλύεται μέσω μαθηματικής βελτιστοποίησης. Πρόσφατα, έχει επιτευχθεί δραματική πρόοδος με τη βαθιά μάθηση, ένα υπο-πεδίο της μηχανικής μάθησης που δημιουργεί μεγάλους παραμετρικούς προσεγγιστές συνθέτοντας απλούστερες συναρτήσεις. Η βαθιά μάθηση υπερέχει όταν εφαρμόζεται σε χώρους μεγάλων διαστάσεων με μεγάλο αριθμό δεδομένων (Bengio Y. et al., 2021).

Ενώ οι περισσότερες επιτυχημένες τεχνικές μηχανικής μάθησης ανήκουν στην οικογένεια της εποπτευόμενης μάθησης, όπου μαθαίνεται η αντιστοίχιση από εισόδους εκπαίδευσης (inputs) σε εκροές (outputs), η εποπτευόμενη μάθηση δεν εφαρμόζεται στα περισσότερα προβλήματα συνδυαστικής βελτιστοποίησης, επειδή δεν έχει πρόσβαση σε βέλτιστες ετικέτες. Ωστόσο, μπορεί κανείς να συγκρίνει την ποιότητα ενός συνόλου λύσεων χρησιμοποιώντας έναν επαληθευτή και να παρέχει κάποιες ανατροφοδοτήσεις ανταμοιβής σε έναν αλγόριθμο εκμάθησης. Ως εκ τούτου,

ακολουθούμε το παράδειγμα της Ενισχυμένης Μάθησης (Reinforcement Learning-RL) για να προσεγγίσουμε τη συνδυαστική βελτιστοποίηση. Εμπειρικά αποδεικνύουμε ότι, ακόμη και όταν χρησιμοποιούνται βέλτιστες λύσεις ως δεδομένα με ετικέτα για τη βελτιστοποίηση μιας εποπτευόμενης χαρτογράφησης, η γενίκευση είναι μάλλον κακή σε σύγκριση με έναν πράκτορα RL που εξερευνά διαφορετικές διαδρομές και παρατηρεί τις αντίστοιχες ανταμοιβές τους (Mazyavkina N. et al.,2021) .

Γενικά, ένας πράκτορας RL δρα στο περιβάλλον μέσω της Διαδικασίας Αποφάσεων Markov (MDP), συλλέγοντας ανταμοιβές και ενημερώνοντας τις μελλοντικές του ενέργειες. Το περιβάλλον αποτελείται από καταστάσεις που σχηματίζουν ένα σύνολο καταστάσεων S, το οποίο θα μπορούσε να είναι είτε διακριτό είτε συνεχές. Για παράδειγμα, μια κατάσταση $s \in S$ μπορεί να περιγραφεί ως η θέση του παίκτη σε κάποιο λαβύρινθο (διακριτή) ή η ροπή που πρέπει να εφαρμοστεί στον κινητήρα (συνεχής). Οι ενέργειες που μπορούν να εκτελέσουν οι πράκτορες δημιουργούν τον χώρο δράσης A και ο κύριος στόχος του πράκτορα είναι να αυξήσει την ανταμοιβή R που λαμβάνει για την εκτέλεση αυτών των ενεργειών. Η συνάρτηση αντιστοίχισης για κάθε κατάσταση s από το S στην καλύτερη αντίστοιχη (η όροι των επιτευχθέντων ανταμοιβών) ενέργεια a από το A ονομάζεται πολιτική, που συνήθως υποδηλώνεται ως $\pi(s)$. Στο κεφάλαιο 2 και 4 θα δούμε αναλυτικά πώς προσεγγίζει η ενισχυμένη μάθηση το TSP.

1.1. Αντικείμενο της Εργασίας

Από την άποψη της συνδυαστικής βελτιστοποίησης, η μηχανική εκμάθηση μπορεί να βοηθήσει στη βελτίωση ενός αλγορίθμου σε μια κατανομή περιπτώσεων προβλημάτων με δύο τρόπους. Από τη μία πλευρά, ο ερευνητής υποθέτει ειδικές γνώσεις σχετικά με τον αλγόριθμο βελτιστοποίησης, αλλά θέλει να αντικαταστήσει ορισμένους απαιτητικούς υπολογισμούς με μια γρήγορη προσέγγιση. Η μάθηση μπορεί να χρησιμοποιηθεί για τη δημιουργία τέτοιων προσεγγίσεων με γενικό τρόπο, δηλαδή, χωρίς την ανάγκη εξαγωγής νέων σαφών αλγορίθμων. Από την άλλη πλευρά, οι ειδικές γνώσεις μπορεί να μην είναι επαρκείς και ορισμένες αλγοριθμικές αποφάσεις μπορεί να μην είναι ικανοποιητικές. Ο στόχος είναι επομένως να εξερευνήσουμε το χώρο αυτών

των αποφάσεων και να μάθουμε από αυτήν την εμπειρία την καλύτερη συμπεριφορά απόδοσης (πολιτική).

Λαμβάνοντας υπόψη τη χρήση μηχανικής μάθησης για την αντιμετώπιση ενός συνδυαστικού προβλήματος, η συνδυαστική βελτιστοποίηση μπορεί να αποσυνθέσει το πρόβλημα σε μικρότερες, ελπίζουμε πιο απλές, μαθησιακές εργασίες. Η συνδυαστική δομή βελτιστοποίησης επομένως λειτουργεί ως σχετικό προηγούμενο για το μοντέλο. Είναι επίσης μια ευκαιρία να αξιοποιηθεί η βιβλιογραφία της συνδυαστικής βελτιστοποίησης, ιδίως όσον αφορά τις θεωρητικές εγγυήσεις (π.χ. σκοπιμότητα και βελτιστοποίηση).

Σημειώνουμε ότι τα CO προβλήματα είναι ένας δημοφιλής τύπος συνδυαστικών προβλημάτων αλλά όχι ο μοναδικός. Άλλα είδη συνδυαστικών προβλημάτων περιλαμβάνουν προβλήματα δημιουργίας, ο κύριος στόχος των οποίων είναι να βρεθούν όλα τα στοιχεία στο σύνολο S που διαθέτουν κάποια ιδιότητα, και προβλήματα απαρίθμησης που εστιάζουν στον υπολογισμό του συνολικού αριθμού στοιχείων ενός συγκεκριμένου τύπου. Για παράδειγμα, η εύρεση όλων των πιθανών αυτομορφισμών γραφήματος, δηλαδή των ισομορφισμών ενός γραφήματος στον εαυτό του, είναι ένα πρόβλημα δημιουργίας, ενώ η εύρεση της πληθώρας της ομάδας αυτομορφισμού είναι ένα πρόβλημα απαρίθμησης. Η επίλυση αυτών των προβλημάτων απαιτεί συχνά την εφαρμογή των αλγορίθμων θεωρητικών προσεγγίσεων, οι οποίοι διαφέρουν ως προς τη φύση τους από τις υπολογιστικές λύσεις, επομένως θα εστιάσουμε μόνο σε προβλήματα συνδυαστικής βελτιστοποίησης σε αυτήν την εργασία.

Το κύριο αντικείμενο αυτής της εργασίας είναι η μελέτη των μεθόδων ενισχυμένης μάθησης (Reinforcement Learning- RL) που έχουν σχεδιαστεί για προβλήματα CO και πιο συγκεκριμένα για το πρόβλημα του Περιοδεύοντος Πωλητή (TSP), ένα από τα πιο σημαντικά και πρακτικά προβλήματα CO. Για να καταλάβουμε καλύτερα το TSP, ας σκεφτούμε έναν πωλητή που ταξιδεύει σε μία περιήγηση σε ένα σύνολο πόλεων. Ο πωλητής πρέπει να επισκεφθεί όλες τις πόλεις ακριβώς μία φορά, ελαχιστοποιώντας τη συνολική διάρκεια της περιοδείας. Το TSP είναι γνωστό ότι είναι ένα πλήρες NP-πρόβλημα (Papadimitriou C., 1977), το οποίο καταγράφει τη δυσκολία εύρεσης αποτελεσματικών λύσεων ακρίβειας σε πολυωνυμικό χρόνο.

Στα επόμενα κεφάλαια θα δούμε αναλυτικά τις μεθόδους RL και ML που επιλέξαμε να αναλύσουμε για την επίλυση του προβλήματος TSP.

1.2. Δομή της Εργασίας

Αφού ολοκληρωθεί το εισαγωγικό κεφάλαιο της εργασίας, στο κεφάλαιο 2 θα πραγματοποιηθεί μία επισκόπηση των μεθόδων Μηχανικής Μάθησης καθώς και τις εφαρμογές τους. Θα παρουσιαστούν επίσης με λεπτομέρεια οι μέθοδοι βαθιάς μάθησης (DL) καθώς και ενισχυμένης μάθησης (RL) που θα μας απασχολήσουν κυρίως στα πλαίσια υλοποίησης της παρούσας εργασίας.

Το κεφάλαιο 3, θα είναι αφιερωμένο στο πρόβλημα του Περιοδεύοντος Πωλητή, γνωστό και ως Travelling Salesman Problem (TSP). Θα δοθεί εκτενής παρουσίαση της φύσης του προβλήματος καθώς και ανασκόπηση των πιο γνωστών μεθόδων επίλυσής του.

Το κεφάλαιο 4, θα προσεγγίσει τα προβλήματα CO που επιλύονται με μεθόδους RL καθώς και παρουσιαστεί μία κατηγοριοποίηση των μεθόδων RL σύμφωνα με τη φύση του προβλήματος που καλούμαστε να επιλύσουμε.

Το κεφάλαιο 5, θα παρουσιάζει μία ανασκόπηση 4 ερευνητικών εργασιών που θεωρήσαμε ότι αποτελούν τις πιο καινοτόμες προσεγγίσεις επίλυσης του TSP με μεθόδους RL. Τέλος, θα συγκρίνουμε τις παραπάνω μεθόδους.

Το κεφάλαιο 6 το οποίο αποτελεί και τον κεντρικό πυλώνα της εργασίας μας, θα παρουσιάσει την προσομοίωση 3 προσεγγίσεων επίλυσης του TSP με μεθόδους RL καθώς πραγματοποιήσαμε αναπαραγωγή των μεθόδων που προαναφέραμε για να μπορούμε να έχουμε μία καλύτερη οπτική από τεχνική άποψη. Επιπρόσθετα, θα περιλαμβάνει και μία δική μας υλοποίηση που θα εμπεριέχει μόνο μεθόδους DL και συγκεκριμένα νευρωνικά δίκτυα.

Καταλήγοντας, το κεφάλαιο 7, θα εμπεριέχει τα συνολικά μας συμπεράσματα καθώς και μία συζήτηση για το πώς θα μπορούσε να εξελιχθεί η μελλοντική έρευνα στο τρέχον επιστημονικό τομέα.

ΚΕΦΑΛΑΙΟ 2

2. Μηχανική Μάθηση και Εφαρμογές

Η μάθηση είναι το κύριο χαρακτηριστικό της ανθρώπινης νοημοσύνης και το βασικό μέσο απόκτησης γνώσης. Η μηχανική μάθηση είναι ο θεμελιώδης τρόπος για να γίνει ο υπολογιστής έξυπνος. Ο R.Shank έχει πει: «Αν ένας υπολογιστής δεν μπορεί να μάθει, δεν θα ονομάζεται έξυπνος». Δεδομένου ότι η μάθηση είναι μια ολοκληρωμένη διανοητική δραστηριότητα που συνδέεται με τη μνήμη, τη σκέψη, την αντίληψη, το συναίσθημα και άλλες νοητικές δραστηριότητες που συνδέονται στενά. Έτσι, ερευνητές από διαφορετικούς τομείς δίνουν αντιστοίχως και τη δική τους διαφορετική ερμηνεία.

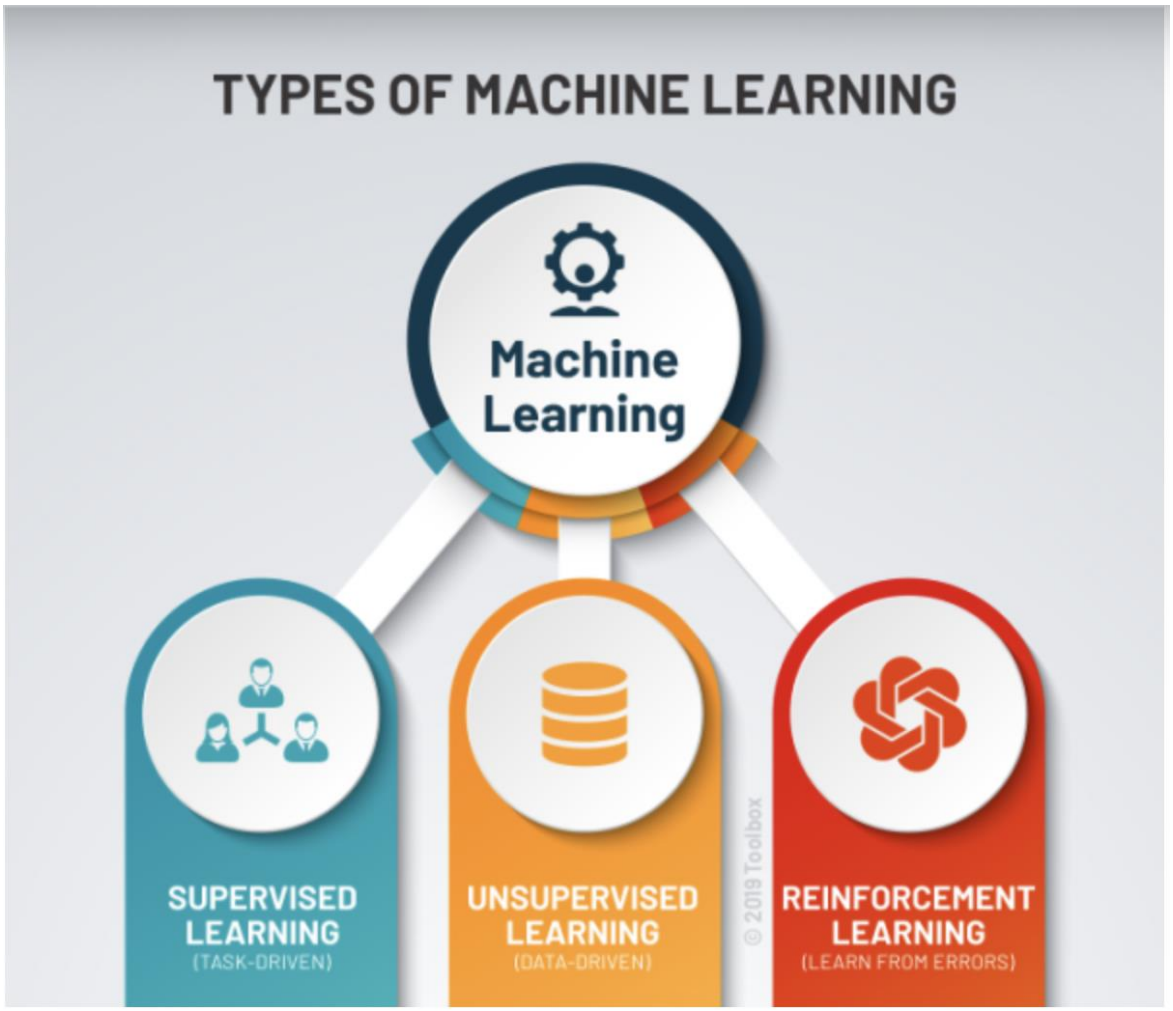
Η μηχανική μάθηση είναι ένα θέμα που μελετά τον τρόπο χρήσης των υπολογιστών για την προσομοίωση των ανθρώπινων μαθησιακών δραστηριοτήτων και τη μελέτη μεθόδων αυτοβελτίωσης των υπολογιστών για την απόκτηση νέων γνώσεων και νέων δεξιοτήτων, τον εντοπισμό υπάρχουσας γνώσης και τη συνεχή βελτίωση της απόδοσης και των επιτευγμάτων τους. Σε σύγκριση με την ανθρώπινη μάθηση, η μηχανική μάθηση μαθαίνει πιο γρήγορα, η συσσώρευση γνώσης διευκολύνει περισσότερο τα αποτελέσματα της μάθησης να διαδίδονται ευκολότερα. Έτσι, οποιαδήποτε πρόοδος του ανθρώπου στον τομέα της μηχανικής μάθησης, θα ενισχύσει την ικανότητα των υπολογιστών, άρα θα έχει αντίκτυπο στην ανθρώπινη κοινωνία.

Είναι γνωστό ότι η τεχνολογία μηχανικής μάθησης έχει χρησιμοποιηθεί ευρέως στο μάρκετινγκ, τα οικονομικά, τις τηλεπικοινωνίες και την ανάλυση δικτύων. Επιπλέον, η μηχανική μάθηση εφαρμόζεται στον τομέα της εξόρυξης δεδομένων σε συνδυασμό και με άλλες εφαρμογές. Οι τυπικές μέθοδοι βασίζονται στην αρχικοποίηση νευρωνικών δικτύων, στην εφαρμογή του εξελικτικού υπολογισμού στην έρευνα μηχανικής μάθησης, στη μελέτη της ταξινόμησης επιπέδου της μηχανικής μάθησης και στη μηχανική μάθηση με βάση το πρόχειρο σύνολο και ούτω καθεξής (Wang H. et al., 2009).

Μπορούμε να χωρίσουμε τους αλγόριθμους μηχανικής μάθησης σε τρεις κύριες ομάδες με βάση τον σκοπό τους:

1. Επιβλεπόμενη Μάθηση (Supervised Learning).
2. Μη επιβλεπόμενη Μάθηση (Unsupervised Learning).

3. Ενισχυμένη Μάθηση (Reinforcement Learning).



Εικόνα 1: Τύποι Μηχανικής Μάθησης[1]

Η επιβλεπόμενη μάθηση είναι ένας από τους πιο βασικούς τύπους μηχανικής μάθησης. Σε αυτόν τον τύπο, ο αλγόριθμος μηχανικής μάθησης εκπαιδεύεται σε δεδομένα με μία συγκεκριμένη ετικέτα. Παρόλο που τα δεδομένα πρέπει να επισημαίνονται με ακρίβεια για να λειτουργήσει αυτή η μέθοδος, η επιβλεπόμενη μάθηση είναι εξαιρετικά ισχυρή όταν χρησιμοποιείται στις σωστές συνθήκες.

Η μη επιβλεπόμενη μάθηση έχει το πλεονέκτημα της δυνατότητας εργασίας με δεδομένα χωρίς ετικέτα. Αυτό σημαίνει ότι δεν απαιτείται ανθρώπινη εργασία για να γίνει το σύνολο δεδομένων αναγνώσιμο από μηχανή, επιτρέποντας την επεξεργασία πολύ μεγαλύτερων συνόλων δεδομένων από το πρόγραμμα.

[1] https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/

Η ενισχυμένη μάθηση εμπνέεται άμεσα από το πώς μαθαίνουν τα ανθρώπινα όντα από τα δεδομένα στη ζωή τους. Διαθέτει έναν αλγόριθμο που βελτιώνεται μόνος του και μαθαίνει από νέες καταστάσεις χρησιμοποιώντας μια μέθοδο δοκιμής και λάθους. Τα ευνοϊκά αποτελέσματα ενθαρρύνονται ή «ενισχύονται» και τα μη ευνοϊκά αποτελέσματα αποθαρρύνονται ή «τιμωρούνται».

2.1. Βαθιά Μάθηση (Deep Learning)

Η βαθιά μάθηση είναι ένας τύπος μηχανικής μάθησης και τεχνητής νοημοσύνης (AI) που μιμείται τον τρόπο με τον οποίο οι άνθρωποι αποκτούν ορισμένους τύπους γνώσης. Η βαθιά μάθηση είναι ένα σημαντικό στοιχείο της επιστήμης δεδομένων, η οποία περιλαμβάνει στατιστικά και προγνωστικά μοντέλα. Είναι εξαιρετικά επωφελές για τους επιστήμονες δεδομένων που είναι επιφορτισμένοι με τη συλλογή, την ανάλυση και την ερμηνεία μεγάλων ποσοτήτων δεδομένων. Η βαθιά μάθηση κάνει αυτή τη διαδικασία πιο γρήγορη και ευκολότερη.

Στην απλούστερη μορφή της, η βαθιά μάθηση μπορεί να θεωρηθεί ως ένας τρόπος αυτοματοποίησης των προγνωστικών αναλυτικών στοιχείων. Ενώ οι παραδοσιακοί αλγόριθμοι μηχανικής μάθησης είναι γραμμικοί, οι αλγόριθμοι βαθιάς μάθησης στοιβάζονται σε μια ιεραρχία αυξανόμενης πολυπλοκότητας και αφαιρετικής πολιτικής[2].

Ας υπογραμμίσουμε ότι, η βαθιά μάθηση είναι ένα υποσύνολο της μηχανικής μάθησης που διαφοροποιείται μέσω του τρόπου με τον οποίο επιλύει προβλήματα. Η μηχανική μάθηση χρειάζεται εξειδίκευση σε ένα συγκεκριμένο τομέα για να επιτύχει τον εντοπισμό των περισσότερων εφαρμοζόμενων λειτουργιών. Από την άλλη πλευρά, η βαθιά μάθηση κατανοεί τα χαρακτηριστικά σταδιακά, εξαλείφοντας έτσι την ανάγκη για εξειδίκευση στον τομέα. Αυτό κάνει τους αλγόριθμους βαθιάς μάθησης να χρειάζονται πολύ περισσότερο χρόνο για να εκπαιδευτούν από τους αλγόριθμους μηχανικής μάθησης, που χρειάζονται μόνο λίγα δευτερόλεπτα έως λίγες ώρες. Ωστόσο, το αντίστροφο ισχύει κατά τη διάρκεια της δοκιμής. Οι αλγόριθμοι βαθιάς μάθησης χρειάζονται πολύ λιγότερο χρόνο για την εκτέλεση δοκιμών από τους αλγόριθμους

[2] https://searchenterpriseai.techtarget.com/definition/deep-learning-deep-neural-network

μηχανικής μάθησης, των οποίων ο χρόνος δοκιμής αυξάνεται μαζί με το μέγεθος των δεδομένων.

Κάθε αλγόριθμος στην ιεραρχία εφαρμόζει έναν μη γραμμικό μετασχηματισμό στην είσοδο του και χρησιμοποιεί αυτό που μαθαίνει για να δημιουργήσει ένα στατιστικό μοντέλο ως έξοδο. Οι επαναλήψεις συνεχίζονται έως ότου η έξοδος φτάσει σε ένα αποδεκτό επίπεδο ακρίβειας. Ο αριθμός των επιπέδων επεξεργασίας μέσω των οποίων πρέπει να περάσουν τα δεδομένα είναι αυτό που ενέπνευσε την εκάστοτε ετικέτα. Για να επιτευχθεί ένα αποδεκτό επίπεδο ακρίβειας, τα προγράμματα βαθιάς μάθησης απαιτούν πρόσβαση σε τεράστιες ποσότητες δεδομένων εκπαίδευσης και επεξεργαστικής ισχύος, κανένα από τα οποία δεν ήταν εύκολα διαθέσιμα στους προγραμματιστές μέχρι την εποχή των μεγάλων δεδομένων και του cloud computing. Επειδή ο προγραμματισμός βαθιάς μάθησης μπορεί να δημιουργήσει πολύπλοκα στατιστικά μοντέλα απευθείας από τη δική του επαναληπτική έξοδο, είναι σε θέση να δημιουργήσει ακριβή μοντέλα πρόβλεψης από μεγάλες ποσότητες μη επισημασμένων, μη δομημένων δεδομένων.

2.1.1. Νευρωνικά Δίκτυα

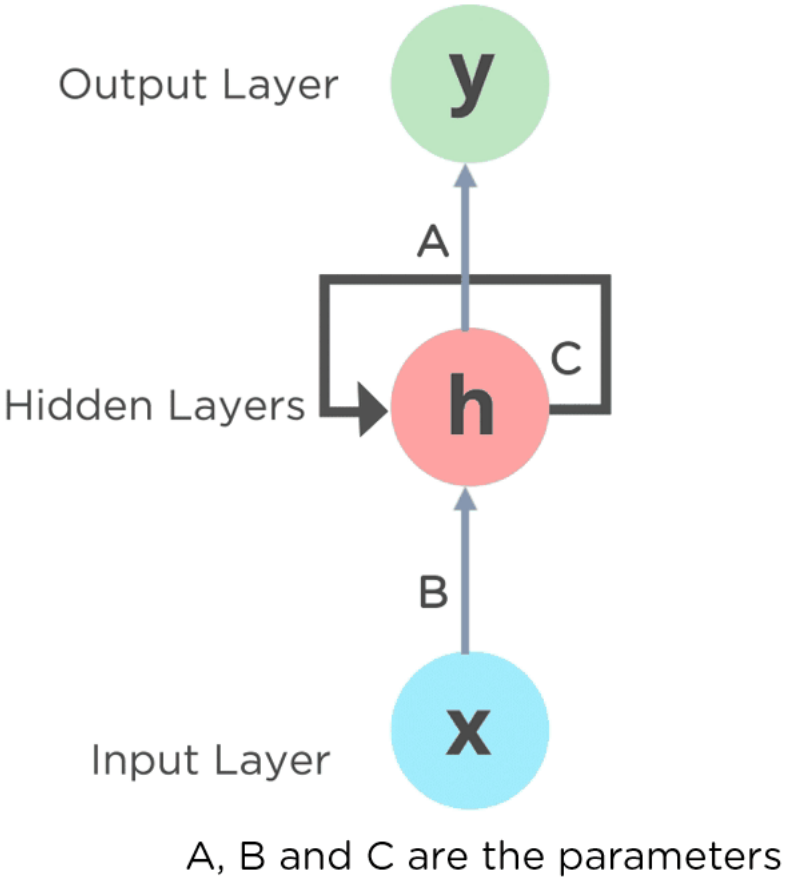
Ένας τύπος προηγμένου αλγόριθμου μηχανικής μάθησης, γνωστός ως τεχνητό νευρωνικό δίκτυο, στηρίζει τα περισσότερα μοντέλα βαθιάς μάθησης. Ως αποτέλεσμα, η βαθιά μάθηση μπορεί μερικές φορές να αναφέρεται ως βαθιά νευρωνική μάθηση ή βαθιά νευρωνική δικτύωση.

Τα νευρωνικά δίκτυα διατίθενται σε πολλές διαφορετικές μορφές, συμπεριλαμβανομένων των επαναλαμβανόμενων - αναδρομικών νευρωνικών δικτύων (RNN), των συνελικτικών νευρωνικών δικτύων (CNN), των τεχνητών νευρωνικών δικτύων (ANN) και των νευρωνικών δικτύων τροφοδοσίας (FNN), και το καθένα έχει οφέλη για συγκεκριμένες περιπτώσεις χρήσης. Ωστόσο, όλα λειτουργούν με κάπως παρόμοιους τρόπους, τροφοδοτώντας δεδομένα και αφήνοντας το μοντέλο να καταλάβει μόνο του εάν έχει λάβει τη σωστή ερμηνεία ή απόφαση σχετικά με ένα δεδομένο στοιχείο δεδομένων.

Τα νευρωνικά δίκτυα περιλαμβάνουν μια διαδικασία δοκιμής και λάθους, επομένως χρειάζονται τεράστιες ποσότητες δεδομένων για την εκπαίδευση. Δεν είναι τυχαίο ότι τα

νευρωνικά δίκτυα έγιναν δημοφιλή μόνο αφού οι περισσότερες επιχειρήσεις αγκάλιασαν την ανάλυση μεγάλων δεδομένων και συσσώρευσαν μεγάλες αποθήκες δεδομένων. Επειδή οι πρώτες επαναλήψεις του μοντέλου περιλαμβάνουν εικασίες σχετικά με το περιεχόμενο μιας εικόνας ή μέρη της ομιλίας, τα δεδομένα που χρησιμοποιούνται κατά το στάδιο εκπαίδευσης πρέπει να φέρουν ετικέτα, ώστε το μοντέλο να μπορεί να δει αν η εικασία του ήταν ακριβής. Αυτό σημαίνει ότι, αν και πολλές επιχειρήσεις που χρησιμοποιούν μεγάλα δεδομένα έχουν μεγάλο όγκο δεδομένων, τα μη δομημένα δεδομένα είναι λιγότερο χρήσιμα. Τα μη δομημένα δεδομένα μπορούν να αναλυθούν από ένα μοντέλο βαθιάς μάθησης μόνο αφού έχουν εκπαιδευτεί και φτάσουν σε ένα αποδεκτό επίπεδο ακρίβειας, αλλά τα μοντέλα βαθιάς μάθησης δεν μπορούν να εκπαιδευτούν σε μη δομημένα δεδομένα.

Παρακάτω παρατίθεται και ένα στιγμιότυπο που απεικονίζει τη δομή ενός Recurrent Neural Network (RNN) που μας απασχόλησε και στα πλαίσια υλοποίησης της παρούσας εργασίας. Επίσης, εργαστήκαμε και με το μοντέλο Long short-term memory (LSTM) που αποτελεί μία υποκατηγορία των RNN που θα εξηγήσουμε σε επόμενο κεφάλαιο.



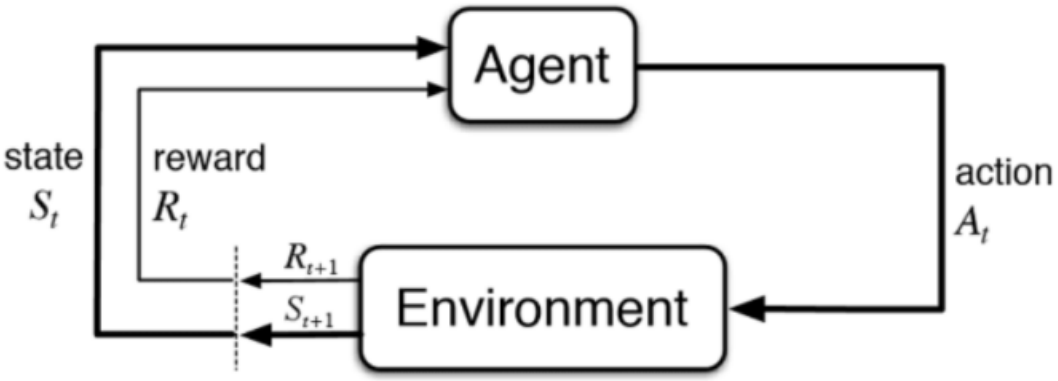
Εικόνα 2: Παράδειγμα RNN

2.2. Ενισχυμένη Μάθηση (Reinforcement Learning)

Όπως αναφέρθηκε παραπάνω, η ενισχυμένη μάθηση (RL) είναι μια τεχνική μηχανικής μάθησης που εστιάζει στην εκπαίδευση ενός αλγόριθμου ακολουθώντας την προσέγγιση "cut-and-try". Ο αλγόριθμος (πράκτορας) αξιολογεί μια τρέχουσα κατάσταση (κατάσταση), αναλαμβάνει μια ενέργεια και λαμβάνει ανατροφοδότηση (ανταμοιβή) από το περιβάλλον μετά από κάθε πράξη. Η θετική ανατροφοδότηση είναι μια ανταμοιβή (με τη συνήθη σημασία της για εμάς), και η αρνητική ανατροφοδότηση είναι τιμωρία για το λάθος.

Ο αλγόριθμος RL μαθαίνει πώς να ενεργεί καλύτερα μέσω πολλών προσπαθειών και αποτυχιών. Η εκμάθηση δοκιμής και σφάλματος συνδέεται με τη λεγόμενη μακροπρόθεσμη ανταμοιβή. Αυτή η ανταμοιβή είναι ο τελικός στόχος που μαθαίνει ο πράκτορας ενώ αλληλοεπιδρά με ένα περιβάλλον μέσω πολυάριθμων δοκιμών και

λαθών. Ο αλγόριθμος λαμβάνει βραχυπρόθεσμες ανταμοιβές που μαζί οδηγούν στη σωρευτική, μακροπρόθεσμη. Έτσι, ο βασικός στόχος της ενισχυμένης μάθησης που χρησιμοποιείται σήμερα είναι να ορίσει την καλύτερη σειρά αποφάσεων που επιτρέπουν στον πράκτορα να λύσει ένα πρόβλημα μεγιστοποιώντας ταυτόχρονα μια μακροπρόθεσμη ανταμοιβή. Και αυτό το σύνολο συνεκτικών ενεργειών μαθαίνεται μέσω της αλληλεπίδρασης με το περιβάλλον και της παρατήρησης των ανταμοιβών σε κάθε κατάσταση (Εικόνα 3).



Εικόνα 3: Διαδικασία RL

2.2.1. Διαφορές μεταξύ ενισχυμένης μάθησης, επιβλεπόμενης μάθησης και μάθησης χωρίς επίβλεψη

Η ενισχυμένη μάθηση διακρίνεται από άλλα στυλ εκπαίδευσης, συμπεριλαμβανομένης της μάθησης με επίβλεψη και χωρίς επίβλεψη, από τον στόχο της και, κατά συνέπεια, τη μαθησιακή προσέγγιση (Εικόνα 4).

Ενισχυμένη Μάθηση VS Επιβλεπόμενης Μάθησης

Στην επιβλεπόμενη μάθηση, ένας πράκτορας «γνωρίζει» ποια εργασία να εκτελέσει και ποιο σύνολο ενεργειών είναι σωστό. Οι επιστήμονες δεδομένων εκπαιδεύουν τον πράκτορα σε ιστορικά δεδομένα με μεταβλητές-στόχους (επιθυμητές απαντήσεις με προγνωστική ανάλυση) δηλαδή δεδομένα με επισήμανση. Ο πράκτορας λαμβάνει άμεση ανατροφοδότηση. Ως αποτέλεσμα της εκπαίδευσης, ένας πράκτορας μπορεί να προβλέψει εάν θα υπάρχουν μεταβλητές στόχου σε νέα δεδομένα ή όχι. Η

εποπτευόμενη μάθηση επιτρέπει την επίλυση εργασιών ταξινόμησης και παλινδρόμησης.

Η ενισχυμένη μάθηση δεν βασίζεται σε επισημασμένα σύνολα δεδομένων: Ο πράκτορας δεν ενημερώνεται ποιες ενέργειες πρέπει να κάνει ή για το βέλτιστο τρόπο εκτέλεσης μιας εργασίας. Η μέθοδος RL χρησιμοποιεί ανταμοιβές και ποινές αντί για ετικέτες που σχετίζονται με κάθε απόφαση σε σύνολα δεδομένων για να υποδείξει εάν μια ενέργεια που πραγματοποιήθηκε είναι καλή ή κακή. Έτσι, ο πράκτορας λαμβάνει σχόλια μόνο αφού ολοκληρώσει την εργασία. Αυτό αποδεικνύει πώς η ανατροφοδότηση με καθυστέρηση και η αρχή δοκιμής και σφάλματος διαφοροποιούν την ενισχυμένη μάθηση από την εποπτευόμενη μάθηση.

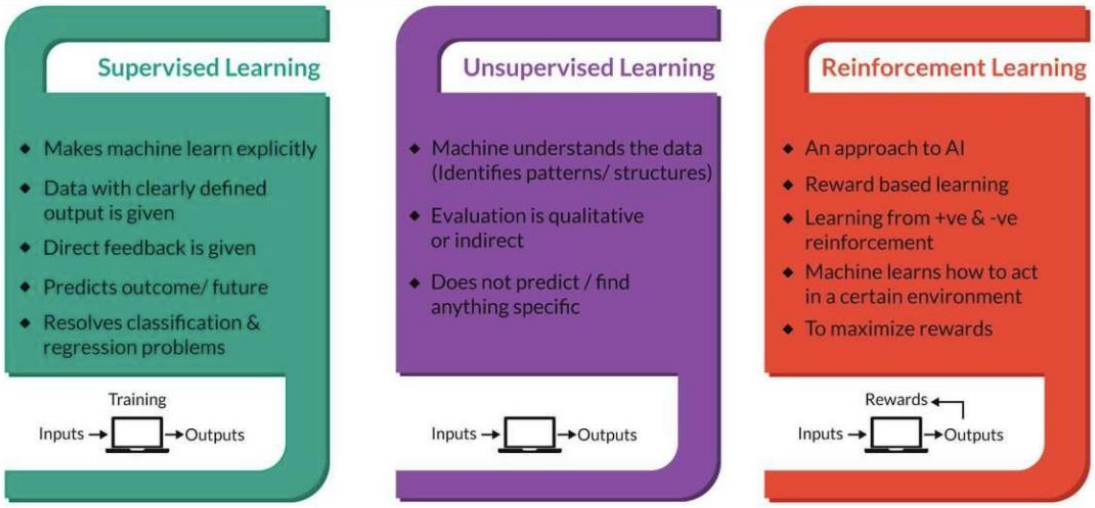
Δεδομένου ότι ένας από τους στόχους του RL είναι να βρει ένα σύνολο διαδοχικών ενεργειών που μεγιστοποιούν μια ανταμοιβή, η διαδοχική λήψη αποφάσεων είναι μια άλλη σημαντική διαφορά μεταξύ αυτών των στυλ εκπαίδευσης αλγορίθμων. Η απόφαση κάθε πράκτορα μπορεί να επηρεάσει τις μελλοντικές του ενέργειες.

Ενισχυμένη Μάθηση VS Μη Επιβλεπόμενη Μάθηση

Στην μάθηση χωρίς επίβλεψη, ο αλγόριθμος αναλύει δεδομένα χωρίς ετικέτα για να βρει κρυφές διασυνδέσεις μεταξύ σημείων δεδομένων και τα δομεί με ομοιότητες ή διαφορές. Η μέθοδος RL στοχεύει στον καθορισμό του καλύτερου μοντέλου δράσης για να λάβει τη μεγαλύτερη μακροπρόθεσμη ανταμοιβή, διαφοροποιώντας το από τη μάθηση χωρίς επίβλεψη όσον αφορά τον βασικό στόχο.

Ενισχυμένη & Βαθιά Μάθηση

Οι περισσότερες από τις υλοποιήσεις ενισχυμένης μάθησης χρησιμοποιούν βαθιά μοντέλα μάθησης. Περιλαμβάνουν τη χρήση νευρωνικών δικτύων ως βασική μέθοδο για την εκπαίδευση πρακτόρων. Σε αντίθεση με άλλες μεθόδους μηχανικής μάθησης, η βαθιά μάθηση ταιριάζει καλύτερα στην αναγνώριση πολύπλοκων μοτίβων σε εικόνες, ήχους και κείμενα. Επιπλέον, τα νευρωνικά δίκτυα επιτρέπουν στους επιστήμονες δεδομένων να ενσωματώνουν όλες τις διαδικασίες σε ένα μόνο μοντέλο χωρίς να διασπούν την αρχιτεκτονική του πράκτορα σε πολλαπλές ενότητες.



Εικόνα 4: Διαφορές μεταξύ επιβλεπόμενης, μη επιβλεπόμενης και ενισχυμένης μάθησης[3]

[3] https://www.altexsoft.com/blog/datascience/reinforcement-learning-explained-overview-comparisons-and-applications-in-business/

ΚΕΦΑΛΑΙΟ 3

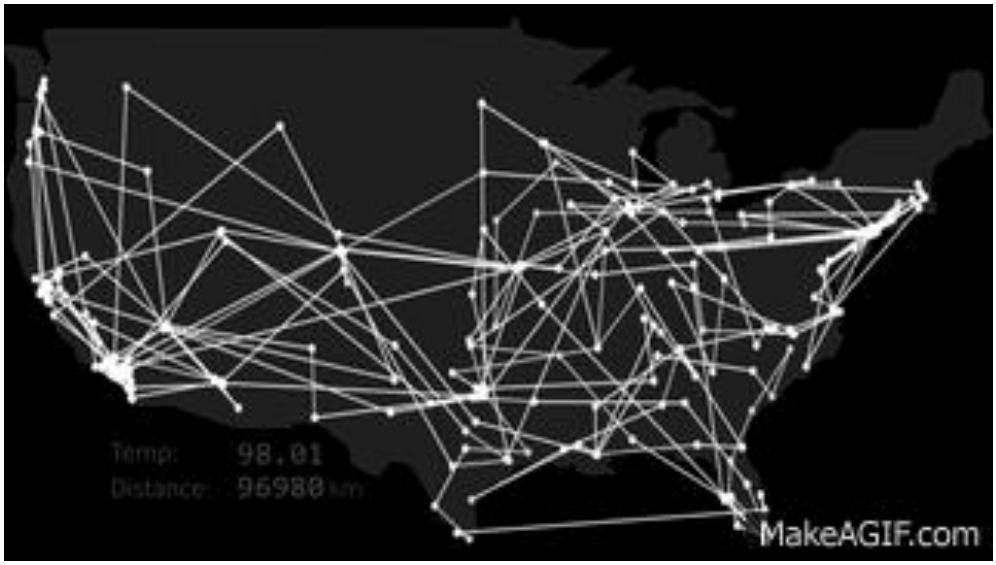
3. Το πρόβλημα του Περιοδεύοντος Πωλητή (TSP)

Σε πάρα πολλές εργασίες που συναντούμε καθημερινά, είναι κρίσιμη η εύρεση της συντομότερης διαδρομής μέσα από μια συλλογή σημείων. Το πιο συνηθισμένο παράδειγμα είναι η παράδοση πακέτων. Η συντομότερη διαδρομή επιλέγεται για εξοικονόμηση καυσίμων και κόστους εργασίας. Άλλα παραδείγματα αποτελούν η κίνηση σε ηλεκτρονικά κυκλώματα και η παραλαβή παραγγελιών από μια αποθήκη και γενικώς στα περισσότερα προβλήματα εφοδιαστικής αλυσίδας (Junger M. et al., 1994).

Στην επιχειρησιακή έρευνα, αυτός ο τύπος προβλήματος ονομάζεται Πρόβλημα του Περιοδεύοντος Πωλητή (TSP). Το πρόβλημα δηλώνει ότι ένας πωλητής πρέπει να επισκεφτεί ένα σύνολο πόλεων μόνο μία φορά, χρησιμοποιώντας τη διαδρομή εντός της συντομότερης απόστασης. Η επίλυση του TSP είναι αρκετά απλή, αλλά επειδή αυτό το πρόβλημα ανήκει στην κατηγορία των NP-complete προβλημάτων, είναι μάλλον δύσκολο να λυθεί.

Τα προβλήματα NP-complete δεν μπορούν να λυθούν σε πολυωνυμικό χρόνο, αλλά μπορούν να επαληθευτούν σε πολυωνυμικό χρόνο. Λόγω αυτής της ιδιότητας το πρόβλημα συνήθως επιλύεται με μεθόδους που δίνουν μια βέλτιστη υπο-λύση σε εύλογο χρόνο(Bose N. K. et al., 1996).

Στην παρούσα μεταπτυχιακή εργασία, θα μας απασχολήσει το 2D Euclidean TSP (Εικ. 5). Κάθε $city_i$ περιγράφεται από τις 2D συντεταγμένες του (x_i,y_i) σε έναν Ευκλείδειο χώρο. Οι περισσότερες προσεγγίσεις χρησιμοποιούν συνήθως την ανάλυση κύριου στοιχείου (PCA) στις κεντρικές συντεταγμένες εισόδου για να αξιοποιήσουν τη χωρική διακύμανση μέσω της εναλλαγής όλων των πόλεων(σημείων). Με αυτόν τον τρόπο, η ευρετική που έχει μαθαίνει το μοντέλο μας, δεν εξαρτάται από τον προσανατολισμό της εισόδου $s = ((x_i,y_i))i∈[1,n]$.



Εικόνα 5: Παράδειγμα TSP

3.1. Ανασκόπηση μεθόδων επίλυσης του TSP

Το πρόβλημα του περιοδεύοντος πωλητή είναι ένα καλά μελετημένο πρόβλημα συνδυαστικής βελτιστοποίησης και έχουν προταθεί πολλοί ακριβείς ή κατά προσέγγιση αλγόριθμοι τόσο για Ευκλείδεια όσο και για μη Ευκλείδεια γραφήματα. Ο Χριστοφίδης (Christofides N., 1976) προτείνει έναν ευρετικό αλγόριθμο που περιλαμβάνει τον υπολογισμό ενός minimum spanning tree και μιας ακριβής αντιστοίχισης ελάχιστου βάρους. Ο αλγόριθμος έχει πολυωνυμικό χρόνο εκτέλεσης.

Ο πιο γνωστός αλγόριθμος δυναμικού προγραμματισμού που παρέχει ακριβείς λύσεις για το TSP έχει πολυπλοκότητα Θ(2nn2), καθιστώντας αδύνατη την κλιμάκωση σε μεγάλες περιπτώσεις, ας πούμε με 50 σημεία. Ωστόσο, οι TSP solvers τελευταίας τεχνολογίας, χάρη σε ευρετικές κατασκευασμένες from scratch που περιγράφουν τον τρόπο πλοήγησης στο χώρο των εφικτών λύσεων με αποτελεσματικό τρόπο, μπορούν να λύσουν συμμετρικές περιπτώσεις TSP με χιλιάδες κόμβους. Το Concorde (Applegate et al., 2006), ευρέως αποδεκτό ως ένας από τους καλύτερους λύτες TSP, κάνει (Dantzig et al., 1954, Padberg & Rinaldi, 1990, Applegate et al., 2003), επαναληπτικά επίλυση των χαλαρώσεων του γραμμικού προγραμματισμού του TSP, σε συνδυασμό με μια προσέγγιση διακλάδωσης και σύνδεσης που κλαδεύει τμήματα του χώρου αναζήτησης που αποδεδειγμένα δεν θα περιέχουν μια βέλτιστη λύση. Ομοίως, η ευρετική Lin

Kernighan-Helsgaun (Helsgaun, 2000), εμπνευσμένη από την ευρετική Lin Kernighan (Lin & Kernighan, 1973), είναι μια τελευταίας τεχνολογίας ευρετική προσέγγιση αναζήτησης για το συμμετρικό TSP και έχει αποδειχθεί ότι λύνει περιπτώσεις με εκατοντάδες κόμβους στη βελτιστοποίηση.

Όπως διαπιστώσαμε, πολλές μέθοδοι έχουν προταθεί για την επίλυση του TSP. Μια επισκόπηση ορισμένων αλγόριθμων με ακριβή επίλυση και κατά προσέγγιση αλγορίθμων για την επίλυση του TSP μπορούν να βρεθούν στο έργο του Laporte (Laporte G, 1992). Μεταξύ αυτών είναι η διατύπωση ακέραιου γραμμικού προγραμματισμού και η μέθοδος branch and bound.

Όπως αναφέρθηκε και παραπάνω, η εύρεση της βέλτιστης λύσης του TSP είναι NP-hard, ακόμη και στη δισδιάστατη Ευκλείδεια περίπτωση (Papadimitriou, 1977), όπου οι κόμβοι είναι δισδιάστατα σημεία και τα βάρη των ακμών είναι ευκλείδειες αποστάσεις μεταξύ ζευγών σημείων. Στην πράξη, οι λύτες TSP βασίζονται σε ευρετικές συναρτήσεις που έχουμε κατασκευάσει οι ίδιοι οι οποίες καθοδηγούν τις διαδικασίες αναζήτησής τους για να βρουν ανταγωνιστικές περιηγήσεις. Παρόλο που αυτές οι ευρετικές λειτουργούν καλά στο TSP, μόλις αλλάξει ελαφρώς το σενάριο του προβλήματος, πρέπει να αναθεωρηθούν. Αντίθετα, οι μέθοδοι μηχανικής μάθησης έχουν τη δυνατότητα να είναι εφαρμόσιμες σε πολλές εργασίες βελτιστοποίησης ανακαλύπτοντας αυτόματα τις δικές τους ευρετικές με βάση τα δεδομένα εκπαίδευσης, απαιτώντας έτσι λιγότερη παρέμβαση από τους τυπικούς solvers που είναι βελτιστοποιημένοι για μία μόνο συγκεκριμένη εργασία και δεν ανταποκρίνονται αποτελεσματικά στη γενίκευση.

Μια άλλη κατηγορία μεθόδων επίλυσης είναι αυτή των νευρωνικών δικτύων. Μια ανασκόπηση των νευρωνικών δικτύων παρουσιάζεται στην εργασία των Altinel et al. (Altinel I. K. et al., 2000). Ένα από τα πρώτα τεχνητά νευρωνικά δίκτυα που σχεδιάστηκαν για την επίλυση του TSP είναι το μοντέλο Hopfield όπως προτείνεται από τους Hopfield και Tank (Hopfiled J. J. and Tank D. W.,1985). Πρόσφατες εργασίες σε αυτή την κατεύθυνση μπορούν να βρεθούν στους Sarwar και Bhatti (Sarwar F. and Bhatti A., 2012). Συγκρίνουν ένα νευρωνικό δίκτυο Hopfield με έναν ευρετικό αλγόριθμο και προτείνουν νέες παραμέτρους για το νευρωνικό δίκτυο Hopfield. Τα αποτελέσματα εξακολουθούν να είναι υποδεέστερα σε σύγκριση με έναν ευρετικό αλγόριθμο. Έρευνες για την υλοποίηση του νευρωνικού δικτύου Hopfield μπορούν να βρεθούν στις εργασίες

των Lau και Widrow (Lau C. and Widrow B., 1990), Woodburn και Murray (Woodburn R. and Murray A., 1997), Luo και Unbehauen (Luo F-L. and Unbehauen R.,1998) και Zhang (Zhang D.,1999).

Ένα άλλο πολύ γνωστό νευρωνικό δίκτυο κατάλληλο για την επίλυση του TSP είναι το νευρωνικό δίκτυο Kohonen (Self Organizing Feature Map). Μια γενική επισκόπηση των εφαρμογών αυτού του τύπου νευρωνικού δικτύου δίνεται στο άρθρο του Kohonen (Kohonen T., 1990). Μια πιο συγκεκριμένη αντιμετώπιση για τη χρήση αυτού του δικτύου στο TSP μπορεί να βρεθεί στην εργασία των Angeniol et al. (Angeniol B. et al., 1988).

Σε αυτό το σημείο θα σχολιάσουμε την επίλυση του TSP μέσω της ενισχυμένης μάθησης. Συνδυαστικά προβλήματα όπως το TSP επιλύονται συχνά διαδοχικά. Συνήθως, μια κατάσταση (state) είναι μια μερική λύση (μια ακολουθία πόλεων που επισκεφθήκαμε) και μια ενέργεια είναι η επόμενη πόλη που θα επισκεφτούμε (μεταξύ αυτών που δεν έχουμε επισκεφτεί ακόμη). Ως απόκριση σε μια ενέργεια, η νέα κατάσταση είναι η ενημερωμένη λύση και το σήμα ανταμοιβής μπορεί είτε να έρθει όταν ολοκληρωθεί μια περιήγηση είτε να είναι σταδιακή. Ένας πράκτορας RL βασίζεται στη δική του εμπειρία δηλαδή ακολουθίες (κατάσταση, δράση, ανταμοιβή, κατάσταση) για να μεγιστοποιήσει τις μελλοντικές ανταμοιβές. Στην πράξη, κάποιος θα μπορούσε είτε να μάθει απευθείας μια (ντετερμινιστική ή στοχαστική) χαρτογράφηση από κατάσταση σε δράση, που ονομάζεται πολιτική π(a|s), είτε να μάθει μια βοηθητική συνάρτηση αξιολόγησης Q (τιμή ή συνάρτηση Q) που μετρά την ποιότητα μιας κατάστασης και χρησιμοποιείται για τη διάκριση μεταξύ των ενεργειών με βάση τη χρησιμότητά τους. Και στις δύο περιπτώσεις, η συνδυαστική δομή του χώρου καταστάσεων S είναι δυσεπίλυτη και απαιτεί τη χρήση συναρτήσεων που θα προσεγγίσουν τη λύση όπως τα βαθιά νευρωνικά δίκτυα.

Το Deep Learning (DL) είναι ένα πλαίσιο γενικής χρήσης για την εκμάθηση αναπαραστάσεων. Με δεδομένο έναν στόχο, ένα νευρωνικό δίκτυο μαθαίνει την αναπαράσταση που απαιτείται για την επίτευξη του στόχου. Τα νευρωνικά δίκτυα υπολογίζουν ιεραρχικές, αφηρημένες αναπαραστάσεις των δεδομένων (μέσω γραμμικών μετασχηματισμών και μη γραμμικών συναρτήσεων ενεργοποίησης) και

μαθαίνουν χαρακτηριστικά χρησιμοποιώντας τον κανόνα της αλυσίδας (Chain Rule) και την Στοχαστική Κάθοδο Κλίσης (Stohastic Gradiend Descent).

ΚΕΦΑΛΑΙΟ 4

4. Προβλήματα Συνδυαστικής Βελτιστοποίησης που επιλύονται με μεθόδους RL και εμβάθυνση στις μεθόδους RL

Οι πιθανοί τρόποι επίλυσης διαφόρων προβλημάτων συνδυαστικής βελτιστοποίησης έχουν ερευνηθεί από επιστήμονες τις τελευταίες δεκαετίες.

Κατά συνέπεια, αυτό οδήγησε στην εμφάνιση μιας τεράστιας οικογένειας μεθόδων επίλυσης προβλημάτων CO, συμπεριλαμβανομένων αυτών που βασίζονται στη μηχανική μάθηση (ML). Η περιεκτική έρευνα των Bengio Y. et al. (2021) πέτυχε να συνοψίσει τους τρόπους προσέγγισης ενός προβλήματος CO από την άποψη της ML καθώς και να απαριθμήσει τις αποτελεσματικές λύσεις σε αυτά τα προβλήματα. Επιπλέον, η εργασία των Zhou J. et al. (2020) η οποία είναι αφιερωμένη στην περιγραφή και τις πιθανές εφαρμογές των Graph Neural Networks (GNN), συνοψίζει την πρόοδο στη διατύπωση προβλημάτων CO από την οπτική γωνία των GNN. Τέλος, οι πιο πρόσφατες έρευνες των Vesselinova N. et al. (2020) και των Guo T. et al. (2019) περιγράφουν τις πιο πρόσφατες προσεγγίσεις ML για την επίλυση των προβλημάτων CO.

Η συγκεκριμένη εργασία επικεντρώνεται αποκλειστικά στο RL ως το ειδικό εργαλείο για την επίλυση διαφόρων προβλημάτων CO, ως εκ τούτου, θα παρέχουμε μια λεπτομερή περιγραφή των μεθόδων RL που χρησιμοποιούνται στις ερευνητικές εργασίες καθώς και το χρήσιμο θεωρητικό υπόβαθρο. Μεταξύ των διαφορετικών προσεγγίσεων, διακρίνουμε τις μεθόδους που βασίζονται σε αξία (value-based), βάσει πολιτικής (policy-based) και τις μεθόδους Neural Monte Carlo Tree Search (MCTS), οι οποίες δε θα μας απασχολήσουν στα πλαίσια της παρούσας εργασίας. Για κάθε μέθοδο, παρέχουμε πρώτα μια θεωρητική εξήγηση που περιγράφει τους κύριους ορισμούς και ιδέες, ακολουθούμενη από τις εφαρμογές κάθε μεθόδου στα συγκεκριμένα προβλήματα CO και κυρίως στο TSP πρόβλημα.

4.1. Επίλυση προβλημάτων Συνδυαστικής Βελτιστοποίησης με Ενισχυμένη Μάθηση

Όλες οι μέθοδοι RL, που καλύπτονται σε αυτήν την εργασία, έχουν ενσωματωθεί με τη Συνδυαστική Βελτιστοποίηση μέσω δύο διαφορετικών παραδειγμάτων: της κύριας και της κοινής μάθησης (principal and joint learning). Στην κύρια μάθηση, ένας πράκτορας λαμβάνει την άμεση απόφαση που αποτελεί μέρος της λύσης ή της ολοκληρωμένης λύσης του προβλήματος και δεν απαιτεί την ανατροφοδότηση από τον λύτη. Για παράδειγμα, στο πρόβλημα TSP, ο πράκτορας μπορεί να παραμετροποιηθεί από ένα νευρωνικό δίκτυο που δημιουργεί σταδιακά μια διαδρομή από ένα σύνολο κορυφών και στη συνέχεια λαμβάνει την ανταμοιβή με τη μορφή του μήκους της κατασκευασμένης διαδρομής, η οποία χρησιμοποιείται για την ενημέρωση της πολιτικής του πράκτορα. Μια άλλη προσέγγιση είναι να μάθουμε την πολιτική του πράκτορα RL στην κοινή εκπαίδευση με ήδη υπάρχοντες λύτες, ώστε να μπορεί να βελτιώσει ορισμένες από τις μετρήσεις για ένα συγκεκριμένο πρόβλημα. Για παράδειγμα, στα προβλήματα MILP μια προσέγγιση που χρησιμοποιείται συνήθως είναι η μέθοδος Branch & Bound, η οποία σε κάθε βήμα επιλέγει έναν κανόνα διακλάδωσης στον κόμβο του δέντρου. Αυτό μπορεί να έχει σημαντικό αντίκτυπο στο συνολικό μέγεθος του δέντρου και, ως εκ τούτου, στο χρόνο εκτέλεσης του αλγορίθμου. Ένας κανόνας διακλάδωσης είναι ένας ευρετικός κανόνας που συνήθως απαιτεί είτε κάποια εξειδίκευση στον τομέα είτε μια διαδικασία συντονισμού των υπερ-παραμέτρων. Ωστόσο, ένας παραμετροποιημένος πράκτορας RL μπορεί να μάθει να μιμείται την πολιτική επιλογής κόμβου λαμβάνοντας ανταμοιβές ανάλογες του χρόνου εκτέλεσης.

4.2. Κατηγοριοποίηση μεθόδων Ενισχυμένης Μάθησης (RL)

Γενικά, οι προσεγγίσεις RL στα προβλήματα της Συνδυαστικής Βελτιστοποίησης μπορούν να θεωρηθούν ως εκμάθηση χρήσιμων ευρετικών, αντί για χειροποίητες που χρησιμοποιούνται συνήθως στην πράξη στην επιχειρησιακή έρευνα. Από αυτή την άποψη, οι μέθοδοι μπορούν να χωριστούν σε εκείνες που μαθαίνουν να κατασκευάζουν ευρετικές και ευρετικές που βελτιώνουν την ήδη υπάρχουσα κατάσταση. Οι μέθοδοι που κατασκευάζουν ευρετικές χτίζουν λύσεις σταδιακά χρησιμοποιώντας μία

εκπαιδευμένη πολιτική που επιλέγει κάθε στοιχείο που θα προστεθεί σε μια μερική λύση. Η δεύτερη ομάδα μεθόδων ξεκινά από κάποια αυθαίρετη λύση και μαθαίνει μια πολιτική που τη βελτιώνει επαναληπτικά.

Η δουλειά μας έχει ως κίνητρο την πρόσφατη επιτυχία στην εφαρμογή των τεχνικών και μεθόδων της ενισχυμένης μάθησης για την επίλυση προβλημάτων CO. Αυτή η εργασία καλύπτει τις πιο πρόσφατες δουλειές που δείχνουν πώς μπορούν να εφαρμοστούν αλγόριθμοι ενισχυμένης μάθησης για την αναδιατύπωση και την επίλυση ορισμένων γνωστών προβλημάτων βελτιστοποίησης, όπως το Πρόβλημα του Περιοδεύοντος Πωλητή (TSP).

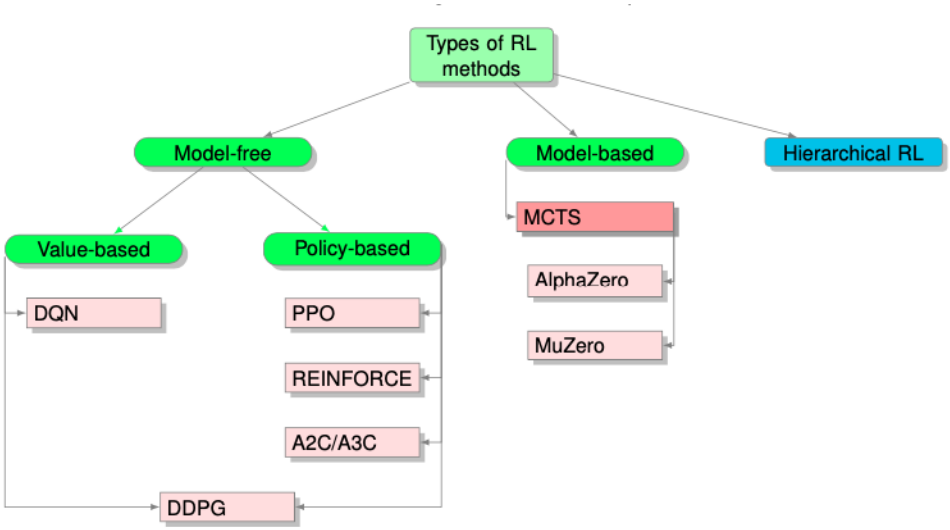
Παρακάτω θα σχολιαστεί η κατηγοριοποίηση των κύριων προσεγγίσεων (Βάσει αξίας, Βάσει Πολιτικής, MCTS) που χρησιμοποιούνται για την επίλυση προβλημάτων CO με RL. Στη συγκεκριμένη εργασία θα μας απασχολήσει μόνο η value-based προσέγγιση και η policy-based. Τα προβλήματα που παρουσιάζονται στον Πίνακα 1, περιλαμβάνουν το μέγιστο ανεξάρτητο σετ (MIS), τη μέγιστη κάλυψη (MC), το μέγιστο κοινό υπογράφημα (MCS), το ελάχιστο κάλυμμα κορυφής (MVC), το πρόβλημα του περιοδεύοντος πωλητή (TSP), το πρόβλημα σακιδίου (KP), το πρόβλημα δρομολόγησης οχήματος (VRP), 3D Πρόβλημα συσκευασίας κάδου (3DBP), χρωματισμός γραφήματος (GC).

Approach	**Problems**
Value	MIS
	MC
	MCS
	MVC
Policy	TSP
	KP
	VRP
	3DBP

Neural MCTS	3DBP
	GC
	MIS

Πίνακας 1: Προσεγγίσεις & Προβλήματα CO

Όπως αναφέρθηκε και παραπάνω, όλες οι υπάρχουσες μέθοδοι RL μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες — μεθόδους που βασίζονται σε μοντέλα και χωρίς μοντέλα. Οι μέθοδοι που βασίζονται σε μοντέλα επικεντρώνονται στα περιβάλλοντα, τα οποία μοντέλα (συναρτήσεις μετάβασης) είναι γνωστά ή είναι δυνατόν να μάθουν και μπορούν να χρησιμοποιηθούν από τον αλγόριθμο κατά τη λήψη των αποφάσεων. Αυτή η ομάδα περιλαμβάνει αλγόριθμους όπως AlphaZero, MuZero, κ.λπ. Από την άλλη πλευρά, οι μέθοδοι χωρίς μοντέλα δεν βασίζονται στη διαθεσιμότητα του μοντέλου του περιβάλλοντος, κάτι που συνήθως συμβαίνει για τα περισσότερα πρακτικά προβλήματα. Οι μέθοδοι χωρίς μοντέλα μπορούν να χωριστούν σε δύο μεγάλες οικογένειες αλγορίθμων RL — μεθόδους που βασίζονται στην πολιτική και σε μεθόδους που βασίζονται σε αξία. Αυτή η κατάτμηση υποκινείται από τον τρόπο εξαγωγής μιας λύσης ενός MDP. Στην περίπτωση των μεθόδων που βασίζονται σε πολιτικές, η πολιτική προσεγγίζεται απευθείας, ενώ οι μέθοδοι που βασίζονται σε αξία εστιάζουν στην προσέγγιση μιας συνάρτησης, η οποία είναι ένα μέτρο της ποιότητας της πολιτικής για κάποιο ζεύγος κατάστασης-ενέργειας στο δεδομένο περιβάλλον.



Εικόνα 6: Τύποι RL

4.2.1. Value-Based Μέθοδοι

Η κύρια εστίαση των μεθόδων μάθησης ενίσχυσης βάσει αξίας είναι η εύρεση μιας βέλτιστης πολιτικής με την προσέγγιση μιας συνάρτησης βέλτιστης τιμής $V^*(s)$ και μιας συνάρτησης τιμής δράσης $Q^*(s, a)$.

Με άλλα λόγια, η συνάρτηση αξίας μας επιτρέπει να αξιολογήσουμε πόσο πολλά υποσχόμενο, όσον αφορά τις μελλοντικές ανταμοιβές, είναι να είμαστε σε κάποια κατάσταση και να ακολουθούμε κάποια συγκεκριμένη πολιτική μακροπρόθεσμα.

Ταυτόχρονα, μπορούμε να σκεφτούμε την ανταμοιβή και τη συνάρτηση αξίας ως λειτουργίες που εξαρτώνται όχι μόνο από την κατάσταση αλλά και από τη δράση. Με αυτόν τον τρόπο, μπορούν να εισαγάγουν τη συνάρτηση τιμής κατάστασης δράσης $Q(s, a)$.

Το Q-learning είναι μια πολύ ισχυρή μέθοδος επίλυσης εργασιών MDP. Ωστόσο, είναι αλήθεια ότι για πολλά προβλήματα του πραγματικού κόσμου είναι πολύ πιο δύσκολο να βρεθεί μια βέλτιστη λύση, ειδικά όταν οι χώροι κατάστασης και δράσης είναι συνεχείς ή πολύ μεγάλοι για να καλυφθούν. Κατά συνέπεια, μερικές φορές μπορεί να είναι πιο βολικό να προσεγγίσουμε απευθείας τη συνάρτηση Q. Με την άνοδο του Deep Learning, τα νευρωνικά δίκτυα (NN) έχουν αποδείξει ότι επιτυγχάνουν πολύ καλά αποτελέσματα σε διάφορα σύνολα δεδομένων μαθαίνοντας χρήσιμες προσεγγίσεις συναρτήσεων μέσω των εισόδων (inputs) υψηλής διάστασης. Αυτό οδήγησε τους ερευνητές να διερευνήσουν τις δυνατότητες των προσεγγίσεων των συναρτήσεων Q μέσω NN, με αποτέλεσμα την εμφάνιση πολλών άρθρων, συμπεριλαμβανομένων των Deep Q-Networks (DQN). Η DQN μέθοδος μπορεί να μάθει τις πολιτικές απευθείας χρησιμοποιώντας ενισχυμένη μάθηση. Σε πρωταρχικές εργασίες που αφορούν την ενισχυμένη μάθηση, οι συγγραφείς εξερεύνησαν την απόδοση της προτεινόμενης μεθόδου στο σημείο αναφοράς παιχνιδιών ATARI, λαμβάνοντας τα προ-επεξεργασμένα πλαίσια του παιχνιδιού ως είσοδο στο συνελικτικό νευρωνικό δίκτυο (CNN) που ορίζεται από τις παραμέτρους θ. Το δίκτυο εξάγει τις κατά προσέγγιση τιμές Q για κάθε ενέργεια ανάλογα με την τρέχουσα κατάσταση εισόδου.

Όλες οι ερευνητικές δουλειές που αφορούν τις value-based μεθόδους, χρησιμοποιούν την ίδια διατύπωση ενημέρωσης της Q-learning συνάρτησης , δηλαδή

ένα δίκτυο γραφημάτων (embedded graph network) χρησιμοποιείται για την παραμετροποίηση της συνάρτησης Q και σε κάθε βήμα εκπαίδευσης η συνάρτηση απώλειας χρησιμοποιείται για backpropagation. Τα δίκτυα γραφημάτων ανήκουν σε μια ομάδα νευρωνικών δικτύων γραφημάτων structure2vec (S2V), η οποία είναι μια δημοφιλής τεχνική κωδικοποίησης γραφημάτων. Σε πολλές έρευνες, οι συγγραφείς έχουν πειραματιστεί με το δίκτυο προσοχής του γραφήματος καθώς και με το δίκτυο μετασχηματιστών ως τον τρόπο κωδικοποίησης της εκάστοτε εργασίας CO. Ενώ τα περισσότερα άρθρα χρησιμοποιούν τις ίδιες αλγοριθμικές προσεγγίσεις, εξακολουθούν να στοχεύουν στην επίλυση διαφορετικών προβλημάτων. Ως εκ τούτου, η κύρια διαφορά των άρθρων που αναφέρουμε στη βιβλιογραφία μας, είναι η αναπαράσταση κατάστασης, δράσης και ανταμοιβής.

Στην περίπτωση της επίλυσης του προβλήματος του Περιοδεύοντος Πωλητή, μια κοινή προσέγγιση είναι:

- η κατάσταση είναι ένα διάνυσμα ενσωμάτωσης γραφήματος p-διαστάσεων, που αντιπροσωπεύει την τρέχουσα περιήγηση κόμβων στο χρονικό βήμα t
- ενώ η ενέργεια επιλέγει έναν άλλο κόμβο, ο οποίος δεν έχει χρησιμοποιηθεί στην τρέχουσα κατάσταση.
- Η ανταμοιβή ορίζεται ως η διαφορά στις συναρτήσεις κόστους μετά τη μετάβαση από την κατάσταση s στην κατάσταση s' όταν κάνουμε κάποια ενέργεια a: $r(s, a)$ = c(h(s'), G) − c(h(s), G), όπου h είναι η συνάρτηση ενσωμάτωσης γραφήματος των επιμέρους λύσεων s και s', G είναι ολόκληρο το γράφημα, c είναι η συνάρτηση κόστους.

4.2.2. Policy-Based Μέθοδοι

Σε αντίθεση με τις μεθόδους που βασίζονται σε αξία άρα στοχεύουν στην εύρεση μιας βέλτιστης συνάρτησης δράσης-κατάστασης $Q*(s, a)$ και ενεργούν άπληστα σε σχέση με αυτήν για να αποκτήσουν τη βέλτιστη πολιτική $\pi*$, οι μέθοδοι που βασίζονται σε πολιτικές προσπαθούν να βρουν απευθείας τη βέλτιστη πολιτική , που αντιπροσωπεύεται από κάποια παραμετρική συνάρτηση $\pi_\theta*$, παρέχοντας βελτιστοποίηση σε σχέση με τις παραμέτρους πολιτικής θ:

- η μέθοδος συλλέγει εμπειρίες στο περιβάλλον με την τρέχουσα πολιτική και τη βελτιστοποιεί με αυτές τις συλλεγόμενες εμπειρίες.
- Μια συνήθης επέκταση του αλγόριθμου REINFORCE είναι η χρήση μιας τιμής βάσης $b(st)$ που μπορεί να υπολογιστεί μέσω μίας μέσης ανταμοιβής στις δειγματοληπτικές τροχιές ή χρησιμοποιώντας έναν εκτιμητή συνάρτησης των τιμών των παραμέτρων $V\phi(st)$, προκειμένου να μειωθεί η διακύμανση στις εκτιμήσεις της τιμής του στόχου χωρίς να παρατηρηθεί μεροληψία.

Actor-Critic Algorithms

Η οικογένεια των αλγορίθμων Actor-Critic (A2C, A3C) επεκτείνει περαιτέρω το REINFORCE με τη τιμή βάσης (baseline) χρησιμοποιώντας τα μέθοδο bootstrapping. Πιο συγκεκριμένα, ενημερώνει τις εκτιμήσεις κατάστασης-τιμής από τις τιμές των επόμενων καταστάσεων.

Εκτός από τις μεθόδους που βασίζονται σε πολιτικές, θα θέλαμε επίσης να δώσουμε κάποια προσοχή σε έναν εννοιολογικά διαφορετικό τρόπο προσέγγισης ενός προβλήματος RL, που είναι η Ιεραρχική Ενισχυμένη Μάθηση (Hierarchical Reinforcement Learning). Οι περισσότερες από τις οικογένειες μεθόδων RL έχουν ένα συγκεκριμένο μειονέκτημα — για να βρεθεί μια καλή πολιτική, ο αλγόριθμος απαιτεί έναν τεράστιο όγκο επαναλήψεων εκπαίδευσης, που είναι ανάλογος με την πολυπλοκότητα της εργασίας. Η κύρια ιδέα πίσω από το Ιεραρχικό RL που αντλεί περαιτέρω έμπνευση από τη συμπεριφορική βιολογία, είναι να ομαδοποιήσει ορισμένες αλληλουχίες πρωταρχικών ενεργειών σε μια ενιαία μακρο-δράση. Στο Deep Reinforcement Learning, προκειμένου να διευκολυνθεί αυτού του είδους η αφαίρεση ενεργειών, οι περισσότεροι αλγόριθμοι δημιουργούν ένα πλαίσιο «manager-learner». Αυτό το πλαίσιο συνήθως αποτελείται από μία πολιτική ανώτατου επιπέδου, η οποία εκπαιδεύεται να επιλέγει από το σύνολο πολλών επιμέρους πολιτικών, οι οποίες, με τη σειρά τους, εκπαιδεύονται για την επίτευξη ορισμένων συγκεκριμένων υπο-στόχων. Η αναπαράσταση των πρωταρχικών στόχων και των πρωταρχικών υπο-στόχων μπορεί επίσης να ποικίλλει, από τους αυτόματα μαθητευόμενους έως τους χειροποίητους, όπως οι ενδιάμεσες καταστάσεις του περιβάλλοντος. Συνολικά, για ορισμένες εργασίες, το Ιεραρχικό RL συμβάλλει στη σημαντική μείωση του χώρου δράσης καθώς και του αριθμού των επαναλήψεων εκπαίδευσης και στην αύξηση της γενίκευσης.

Μία από τις πρώτες προσπάθειες εφαρμογής αλγορίθμων βάσει πολιτικής σε προβλήματα συνδυαστικής βελτιστοποίησης έχει γίνει για την επίλυση προβλημάτων TSP και Knapsack χρησιμοποιώντας τον αλγόριθμο REINFORCE. Για αυτήν την προσπάθεια, χρησιμοποιείται μια αρχιτεκτονική δικτύου δεικτών (Pointer Network) για την κωδικοποίηση της ακολουθίας εισόδου. Η λύση κατασκευάζεται διαδοχικά από μια κατανομή στην είσοδο χρησιμοποιώντας τον μηχανισμό δείκτη του αποκωδικοποιητή και εκπαιδεύεται παράλληλα. Επιπλέον, προτείνονται διάφορες στρατηγικές συμπερασμάτων για την κατασκευή μιας λύσης. Μαζί με την άπληστη αποκωδικοποίηση (Greedy Decoding) και δειγματοληψία (Sampling), προτείνεται η προσέγγιση Ενεργής Αναζήτησης. Η Ενεργή Αναζήτηση (Active Search) επιτρέπει την εκμάθηση της λύσης για την περίπτωση ενός προβλήματος δοκιμής, είτε ξεκινώντας από ένα εκπαιδευμένο είτε από μη εκπαιδευμένο μοντέλο.

ΚΕΦΑΛΑΙΟ 5

5. Ανάλυση επιλεγμένης βιβλιογραφίας

5.1. Παρουσίαση βιβλιογραφίας και γνωστών υλοποιήσεων

Στο συγκεκριμένο κεφάλαιο θα αναλυθούν 4 προσεγγίσεις που επιλέχθηκαν από τη βιβλιογραφία που μελετήθηκε στα πλαίσια υλοποίησης της παρούσας εργασίας. Αφορούν όλες την επίλυση του προβλήματος TSP με μεθόδους RL. Παρατηρούνται διαφορές στην αρχιτεκτονική των υλοποιήσεων που θα παρουσιαστούν και θα σχολιαστούν παρακάτω.

5.1.1. Προσέγγιση 1

Neural Combinatorial Optimization with Reinforcement Learning

Στο παρόν paper οι Bello et al., (2019) επικεντρώνονται στο πρόβλημα του περιοδεύοντος πωλητή (TSP) και εκπαιδεύουν ένα αναδρομικό νευρωνικό δίκτυο (RNN) που, δεδομένου ενός συνόλου συντεταγμένων διαφόρων πόλεων, προβλέπει μια κατανομή σε διαφορετικές μεταθέσεις πόλεων. Χρησιμοποιώντας το αρνητικό μήκος διαδρομής ως ανταμοιβή, βελτιστοποιούν τις παραμέτρους του αναδρομικού νευρωνικού δικτύου χρησιμοποιώντας μια μέθοδο «policy gradient».

Εξετάζουν δύο προσεγγίσεις που βασίζονται σε policy gradient μέθοδο (Williams, 1992). Η πρώτη προσέγγιση, που ονομάζεται προ-εκπαίδευση RL, χρησιμοποιεί ένα σύνολο εκπαίδευσης για τη βελτιστοποίηση ενός επαναλαμβανόμενου νευρωνικού δικτύου (RNN) που παραμετροποιεί μια στοχαστική πολιτική έναντι των λύσεων, χρησιμοποιώντας την αναμενόμενη ανταμοιβή ως στόχο. Κατά τη δοκιμή, η πολιτική διορθώνεται και λαμβάνουμε τα συμπεράσματά μας μέσω άπληστης αποκωδικοποίησης (greedy decoding) ή δειγματοληψίας (sampling). Η δεύτερη προσέγγιση, που ονομάζεται ενεργή αναζήτηση (Active Search), δεν περιλαμβάνει προκατάρτιση. Ξεκινά από μια τυχαία πολιτική και βελτιστοποιεί επαναληπτικά τις παραμέτρους RNN σε μια μεμονωμένη περίπτωση δοκιμής, χρησιμοποιώντας ξανά τον αναμενόμενο στόχο ανταμοιβής, ενώ παρακολουθεί την καλύτερη λύση που

δειγματολήφθηκε κατά την αναζήτηση. Διαπιστώνουν ότι ο συνδυασμός προ-εκπαίδευσης RL και ενεργής αναζήτησης λειτουργεί καλύτερα στην πράξη.

Το δίκτυο κωδικοποιητή διαβάζει την ακολουθία εισόδου s, μία πόλη τη φορά, και τη μετατρέπει σε μια ακολουθία καταστάσεων λανθάνουσας μνήμης $\{enc_i\}^n$ i=1 όπου $enc_i \in R^d$. Η είσοδος στο δίκτυο κωδικοποιητή στο χρονικό βήμα i είναι μια d-διάστατη ενσωμάτωση ενός δισδιάστατου σημείου x_i, το οποίο λαμβάνεται μέσω ενός γραμμικού μετασχηματισμού του x_i που μοιράζεται σε όλα τα βήματα εισόδου. Το δίκτυο αποκωδικοποιητή διατηρεί επίσης τις καταστάσεις λανθάνουσας μνήμης $\{dec_i\}^n$ i=1 όπου $dec_i \in R^d$ και, σε κάθε βήμα i, χρησιμοποιεί έναν μηχανισμό κατάδειξης για να παράγει μια κατανομή στην επόμενη πόλη που θα επισκεφθείτε στην περιήγηση. Μόλις επιλεγεί η επόμενη πόλη, μεταβιβάζεται ως είσοδος στο επόμενο βήμα του αποκωδικοποιητή. Η είσοδος του πρώτου βήματος αποκωδικοποιητή είναι ένα διάνυσμα d που αντιμετωπίζεται ως εκπαιδεύσιμη παράμετρος του νευρωνικού τους δικτύου.

Η συνάρτηση προσοχής τους λαμβάνει ως είσοδο ένα διάνυσμα ερωτήματος $q = dec_i \in R^d$ και ένα σύνολο διανυσμάτων αναφοράς ref = {enc1, . . . , enck} όπου $enc_i \in R^d$, και προβλέπει μια κατανομή A(ref, q) στο σύνολο των k αναφορών. Αυτή η κατανομή πιθανότητας αντιπροσωπεύει τον βαθμό στον οποίο το μοντέλο δείχνει στην αναφορά r_i όταν βλέπει το ερώτημα q.

Ο στόχος της εκπαίδευσής τους είναι η αναμενόμενη διάρκεια της περιοδείας, δεδομένου ενός γραφήματος εισαγωγής s:

$$J(\theta \mid s) = E_{\pi \sim p\theta(.|s)} L(\pi \mid s)$$

Η χρήση μιας παραμετρικής γραμμής βάσης για την εκτίμηση της αναμενόμενης διάρκειας περιοδείας $E_{\pi \sim p\theta(.|s)} L(\pi \mid s)$ συνήθως βελτιώνει τη μάθηση. Ως εκ τούτου, εισάγουν ένα βοηθητικό δίκτυο, που ονομάζεται κριτικός και παραμετροποιείται με θ_v, για να μάθουν την αναμενόμενη διάρκεια περιήγησης που βρέθηκε από την τρέχουσα πολιτική τους p_θ, δεδομένης μιας ακολουθίας εισόδου s. Ο κριτικός εκπαιδεύεται με στοχαστική κλίση κατάβασης σε έναν στόχο μέσου τετραγώνου σφάλματος μεταξύ των προβλέψεών του $b_{\theta v}$ (s) και της πραγματικής διάρκειας περιοδείας που δειγματοληψία από την πιο πρόσφατη πολιτική.

Critic's architecture for TSP

Σε αυτό το σημείο εξηγούν πώς ο κριτικός τους αντιστοιχίζει μια ακολουθία εισόδου s σε μια βασική πρόβλεψη bθv (s). Ο κριτικός τους περιλαμβάνει τρεις μονάδες νευρωνικών δικτύων: 1) έναν κωδικοποιητή LSTM, 2) ένα μπλοκ διαδικασίας LSTM και 3) έναν αποκωδικοποιητή νευρωνικού δικτύου ReLU 2 επιπέδων. Ο κωδικοποιητής του έχει την ίδια αρχιτεκτονική με αυτή του κωδικοποιητή του δικτύου δεικτών τους και κωδικοποιεί μια ακολουθία εισόδου s σε μια ακολουθία καταστάσεων λανθάνουσας μνήμης και σε μια κρυφή κατάσταση h. Το μπλοκ διεργασίας, όπως (Vinyals et al., 2015a), στη συνέχεια εκτελεί βήματα υπολογισμού P στην κρυφή κατάσταση h. Κάθε βήμα επεξεργασίας ενημερώνει αυτήν την κρυφή κατάσταση ρίχνοντας μια ματιά στις καταστάσεις μνήμης και τροφοδοτεί την έξοδο της συνάρτησης ως είσοδο στο επόμενο βήμα επεξεργασίας. Στο τέλος του μπλοκ διεργασίας, η λαμβανόμενη κρυφή κατάσταση στη συνέχεια αποκωδικοποιείται σε μια πρόβλεψη baseline (δηλαδή σε ένα μόνο βαθμωτό) από δύο πλήρως συνδεδεμένα στρώματα.

Search Strategies

Καθώς η αξιολόγηση της διάρκειας μιας περιοδείας του πωλητή είναι φθηνή υπολογιστικά, ο πράκτορας (agent) του TSP τους μπορεί εύκολα να προσομοιώσει μια διαδικασία αναζήτησης στο χρόνο συμπερασμάτων, εξετάζοντας πολλαπλές υποψήφιες λύσεις ανά γράφημα και επιλέγοντας τις καλύτερες. Αυτή η διαδικασία εξαγωγής συμπερασμάτων μοιάζει με το πώς οι λύτες αναζητούν ένα μεγάλο σύνολο εφικτών λύσεων. Σε αυτό το άρθρο, εξετάζουν δύο στρατηγικές αναζήτησης οι οποίες αναφέρονται ως δειγματοληψία και ενεργή αναζήτηση.

5.1.2. Προσέγγιση 2

Learning Heuristics for the TSP by Policy Gradient

Στην συγκεκριμένη υλοποίηση των Deudon et al. (2019), οι συντεταγμένες της εκάστοτε πόλης χρησιμοποιούνται ως είσοδοι και το νευρωνικό δίκτυο εκπαιδεύεται χρησιμοποιώντας ενισχυμένη μάθηση για την πρόβλεψη μιας κατανομής στις μεταθέσεις πόλεων. Το προτεινόμενο πλαίσιο διαφέρει από την παραπάνω προσέγγιση,

καθώς δεν χρησιμοποιούν την αρχιτεκτονική της Μακροπρόθεσμης-Βραχυπρόθεσμης Μνήμης (LSTM) και επέλεξαν να σχεδιάσουν τον δικό τους κριτικό αλγόριθμο για να υπολογίσουν μια βάση - baseline για τη διάρκεια της περιοδείας που οδηγεί σε πιο αποτελεσματική μάθηση. Επιπλέον ενισχύουν περαιτέρω την προσέγγιση της λύσης με τη γνωστή ευρετική 2-opt.

Κάθε $city_i$ περιγράφεται από τις 2D συντεταγμένες του (x_i,y_i) σε έναν Ευκλείδειο χώρο. Χρησιμοποιούν την ανάλυση κεντρικού στοιχείου (PCA) στις κεντρικές συντεταγμένες εισόδου για να εκμεταλλευτούν την εναλλαγή όλων των πόλεων. Με αυτόν τον τρόπο, η ευρετική που μαθαίνεται δεν εξαρτάται από τον προσανατολισμό της εισόδου $s = ((x_i,y_i))i∈[1,n]$.

TSP Encoder

Ο σκοπός του κωδικοποιητή τους είναι να αποκτήσει μια αναπαράσταση για κάθε δράση (πόλη) δεδομένου του πλαισίου της. Η έξοδος του κωδικοποιητή τους είναι ένα σύνολο διανυσμάτων δράσης $A = (a_1,...,a_n)$, το καθένα αντιπροσωπεύει μια πόλη που αλληλοεπιδρά με άλλες πόλεις. Ο νευρωνικός κωδικοποιητής τους εμπνέεται από τις πρόσφατες εξελίξεις στη Μετάφραση Νευρωνικής Μηχανής. Ο actor και ο critic τους χρησιμοποιούν μηχανισμούς νευρωνικής προσοχής (neural attention mechanisms) για να κωδικοποιήσουν τις πόλεις ως σύνολο (και όχι ως ακολουθία όπως παραπάνω).

Multi-Head Attention

Οι μηχανισμοί νευρωνικής προσοχής επιτρέπουν στα ερωτήματα να αλληλοεπιδρούν με ζεύγη κλειδιών-τιμών. Για το TSP, τα ερωτήματα και τα ζεύγη κλειδιού-τιμής q_i , k_i , v_i $∈ R^d$ λαμβάνονται μετατρέποντας γραμμικά κάθε $city_i ∈ R^d$ και εφαρμόζοντας μια μη γραμμικότητα μέσω της συνάρτησης ReLu (non-linearity).

Decoder

Η αρχιτεκτονική τους νευρωνικού δικτύου χρησιμοποιεί τον κανόνα της αλυσίδας για να παραγοντοποιήσει την πιθανότητα μιας περιήγησης. Σε αντίθεση με τους Bello et al. που συνοψίζει όλες τις προηγούμενες ενέργειες σε ένα διάνυσμα σταθερού μήκους, αυτό το μοντέλο ξεχνά ρητά μετά από K = 3 βήματα, παραλείποντας τα δίκτυα LSTM. Σε

κάθε χρόνο εξόδου t, αντιστοιχίζουν τις τρεις τελευταίες δειγματοληπτικές ενέργειες (επισκέψεις σε πόλεις) στο ακόλουθο διάνυσμα ερωτήματος:

$$q_t = \mathrm{ReLu}(W1a_{\pi(t-1)} + W2a_{\pi(t-2)} + W3a_{\pi(t-3)}) \in R^{d'}$$

Παρόμοια με τους Bello et al., το διάνυσμα ερώτησής τους q_t αλληλοεπιδρά με ένα σύνολο n διανυσμάτων για να ορίσει μια κατανομή κατάδειξης στον χώρο δράσης. Μόλις γίνει δειγματοληψία της επόμενης πόλης, η τροχιά q_{t+1} ενημερώνεται με το επιλεγμένο διάνυσμα δράσης και η διαδικασία τελειώνει όταν ολοκληρωθεί η περιήγηση.

Pointing Mechanism

Χρησιμοποιούν τον ίδιο μηχανισμό κατάδειξης με τους Bello et al. για να προβλέψουν μια κατανομή σε πόλεις με κωδικοποιημένες ενέργειες (πόλεις) και μια αναπαράσταση κατάστασης (διάνυσμα ερωτήματος). Η κατάδειξη σε μια συγκεκριμένη θέση στην ακολουθία εισόδου επιτρέπει την προσαρμογή του ίδιου πλαισίου σε περιηγήσεις μεταβλητού μήκους.

Εκπαιδεύουν το Νευρωνικό τους Δίκτυο χρησιμοποιώντας τον κανόνα μάθησης REINFORCE με έναν critic για να μειώσουν τη διακύμανση των διαβαθμίσεων. Για το TSP, χρησιμοποιούν τη διάρκεια της περιοδείας ως ανταμοιβή $r(\pi|s) = L(\pi|s)$ (την οποία επιδιώκουν να ελαχιστοποιήσουν).

Actor-Critic Algorithm

Μαθαίνουν τις παραμέτρους του actor θ ξεκινώντας από μια τυχαία πολιτική και βελτιστοποιώντας τις επαναληπτικά με τον κανόνα μάθησης REINFORCE και το Stochastic Gradient Descent (SGD), σε περιπτώσεις που δημιουργούνται εν κινήσει. Ο critic τους χρησιμοποιεί τον ίδιο κωδικοποιητή με τον actor τους. Χρησιμοποιεί μια φορά τον μηχανισμό κατάδειξης με $q = 0_d'$. Η κατανομή κατάδειξης του κριτικού στις πόλεις $p_\phi(s)$ ορίζει ένα διάνυσμα αναλαμπής gls που υπολογίζεται ως το σταθμισμένο άθροισμα των διανυσμάτων δράσης $A = (a_1, ..., a_n)$. Ο critic εκπαιδεύεται ελαχιστοποιώντας το Τετράγωνο του Μέσου Σφάλματος μεταξύ των προβλέψεών του και των ανταμοιβών του actor.

5.1.3. Προσέγγιση 3

Combinatorial Optimization by Graph Pointer Networks and Hierarchical Reinforcement Learning (Ma et al., 2019).

Σε αυτή την εργασία, γίνεται χρήση των "Graph Pointer Networks" (GPN) που εκπαιδεύονται με χρήση ενισχυμένης μάθησης (RL) για την αντιμετώπιση του προβλήματος του περιοδεύοντος πωλητή (TSP). Τα GPN βασίζονται σε δίκτυα δεικτών εισάγοντας ένα επίπεδο ενσωμάτωσης γραφήματος στην είσοδο, το οποίο καταγράφει τις σχέσεις μεταξύ των κόμβων. Επιπλέον, για να προσεγγίσουν λύσεις σε περιορισμένα προβλήματα συνδυαστικής βελτιστοποίησης, όπως το TSP με χρονικούς περιορισμούς, εκπαιδεύουν ιεραρχικά GPN (HGPN) χρησιμοποιώντας RL, το οποίο μαθαίνει μια ιεραρχική πολιτική για να βρει μια βέλτιστη μετάθεση πόλης υπό περιορισμούς. Κάθε επίπεδο της ιεραρχίας έχει σχεδιαστεί με ξεχωριστή συνάρτηση ανταμοιβής.

Οι προηγούμενες εργασίες έχουν επιτύχει καλά κατά προσέγγιση αποτελέσματα σε διάφορα προβλήματα συνδυαστικής βελτιστοποίησης, αλλά προβλήματα συνδυαστικής βελτιστοποίησης με περιορισμούς, π.χ. TSP με χρονικό περιορισμό (TSPTW), δεν είχαν ληφθεί πλήρως υπόψη. Για την αντιμετώπιση περιορισμένων προβλημάτων, οι Bello et al. πρότειναν μια μέθοδο ποινής, η οποία πρόσθεσε έναν όρο ποινής για ανέφικτες λύσεις στη συνάρτηση ανταμοιβής. Ωστόσο, η μέθοδος ποινής μπορεί να οδηγήσει σε ασταθή εκπαίδευση και οι υπερ-παράμετροι του όρου ποινής είναι συνήθως δύσκολο να συντονιστούν. Μια καλύτερη επιλογή για εκπαίδευση είναι η χρήση ιεραρχικών μεθόδων RL, οι οποίες έχουν εφαρμοστεί ευρέως για την αντιμετώπιση σύνθετων προβλημάτων όπως βιντεοπαιχνίδια. Το βασικό κίνητρο για την ιεραρχική RL μεθοδολογία είναι ο διαχωρισμός σύνθετων εργασιών σε πολλά απλά υπο-προβλήματα που μαθαίνονται σε μια ιεραρχία. Σε αυτή την εργασία, διερευνούν τη χρήση ιεραρχικών μεθόδων RL για την αντιμετώπιση προβλημάτων συνδυαστικής βελτιστοποίησης με περιορισμούς, οι οποίοι χωρίζονται σε διαφορετικές υπο-εργασίες. Κάθε επίπεδο της ιεραρχίας μαθαίνει να αναζητά τις εφικτές λύσεις υπό περιορισμούς ή μαθαίνει τις ευρετικές για τη βελτιστοποίηση της αντικειμενικής συνάρτησης.

Οι συνεισφορές αυτής της εργασίας είναι τρεις:

- Πρώτον, προτείνουν ένα δίκτυο δείκτη γραφήματος (GPN) για την αντιμετώπιση του TSP. Το GPN επεκτείνει το δίκτυο δεικτών με επίπεδα ενσωμάτωσης γραφημάτων και επιτυγχάνει ταχύτερη σύγκλιση.
- Δεύτερον, προσθέτουν ένα διανυσματικό πλαίσιο στην αρχιτεκτονική GPN και εκπαιδεύουν χρησιμοποιώντας πρόωρη διακοπή προκειμένου να γενικεύσουν το μοντέλο τους για την αντιμετώπιση περιπτώσεων TSP μεγαλύτερης κλίμακας, π.χ. το TSP1000, από ένα μοντέλο εκπαιδευμένο σε πολύ μικρότερο στιγμιότυπο TSP50.
- Τρίτον, χρησιμοποιούν ένα ιεραρχικό πλαίσιο RL μαζί με την αρχιτεκτονική GPN για την αποτελεσματική επίλυση του TSP με χρονικούς περιορισμούς. Για κάθε εργασία, διεξάγουν πειράματα για να συγκρίνουν την απόδοση του μοντέλου τους με τις υπάρχουσες baselines και προηγούμενες εργασίες.

Κάθε επίπεδο αντιστοιχεί σε μια διαφορετική εργασία RL, επομένως οι συναρτήσεις ανταμοιβής είναι σχεδιασμένες ώστε να είναι διαφορετικές για κάθε επίπεδο. Υπάρχουν δύο φυσικοί τρόποι για τη διαμόρφωση προβλημάτων TSP με ιεραρχικό τρόπο. Πρώτον, ορίζουν τις συναρτήσεις ανταμοιβής χαμηλότερου επιπέδου σε απλές λύσεις ώστε να βρίσκονται στο εφικτό σύνολο του προβλήματος βελτιστοποίησης και ορίζουν τις συναρτήσεις ανταμοιβής υψηλότερου επιπέδου ως τον αρχικό στόχο βελτιστοποίησης. Αντίστροφα, κατατάσσουν συναρτήσεις ανταμοιβής με αυξανόμενη δυσκολία βελτιστοποίησης: το πρώτο επίπεδο επιχειρεί να λύσει το απλό TSP, στο δεύτερο στρώμα δίνεται ένα TSP με έναν απλό περιορισμό, και ούτω καθεξής.

GPN Architecture

Το GPN Architecture αποτελείται από ένα στοιχείο κωδικοποιητή και αποκωδικοποιητή (Encoder-Decoder).

Encoder

Ο κωδικοποιητής περιλαμβάνει δύο μέρη: τον κωδικοποιητή σημείων και τον κωδικοποιητή γραφήματος. Για τον κωδικοποιητή σημείων, κάθε συντεταγμένη πόλης x είναι ενσωματωμένη σε ένα διάνυσμα υψηλότερης διάστασης $\tilde{x_i} \in R^d$, όπου d είναι η κρυφή διάσταση. Αυτός ο γραμμικός μετασχηματισμός μοιράζεται τα βάρη σε όλες τις πόλεις x_i. Στη συνέχεια, το διάνυσμα $\tilde{x_i}$ για την τρέχουσα πόλη x_i κωδικοποιείται από ένα

LSTM. Η κρυφή μεταβλητή x_{hi} του LSTM μεταβιβάζεται τόσο στον αποκωδικοποιητή στο τρέχον βήμα όσο και στον κωδικοποιητή στο επόμενο χρονικό βήμα.

Για τον κωδικοποιητή γραφήματος, χρησιμοποιούν στρώματα ενσωμάτωσης γραφήματος για να κωδικοποιήσουν όλες τις συντεταγμένες πόλης $X = [xT_1 , ..., xT_N]^T$ και να τις περάσουν στον αποκωδικοποιητή.

Graph Embedding Layer -Επίπεδο ενσωμάτωσης γραφήματος

Στο TSP, οι πληροφορίες περιβάλλοντος ενός κόμβου-πόλης περιλαμβάνουν τις πληροφορίες των γειτόνων της εκάστοτε πόλης. Σε ένα GPN, οι πληροφορίες περιβάλλοντος λαμβάνονται με την κωδικοποίηση όλων των συντεταγμένων πόλεων X μέσω ενός νευρωνικού δικτύου γραφήματος (GNN).

Vector Context

Αντί να χρησιμοποιούν άμεσα χαρακτηριστικά συντεταγμένων, σε αυτήν την εργασία, χρησιμοποιούν τα διανύσματα που δείχνουν από την τρέχουσα πόλη σε όλες τις άλλες πόλεις ως πλαίσιο, τα οποία αναφέρονται ως διανυσματικό πλαίσιο.

Decoder

Ο αποκωδικοποιητής βασίζεται σε έναν μηχανισμό προσοχής και εξάγει τη διεπαφή χρήστη του διανύσματος δείκτη, η οποία στη συνέχεια περνά σε ένα επίπεδο softmax για να δημιουργήσει μια κατανομή στις επόμενες υποψήφιες πόλεις.

Policy

Η πολιτική διανομής σε όλες τις υποψήφιες πόλεις δίνεται από:

$$\pi\theta(a_i|s_i) = p_i = \text{softmax}(u_i)$$

Προβλέπουν την επόμενη πόλη που θα επισκεφτούν $a_i = x_{\sigma(i+1)}$, δειγματοληπτικά ή επιλέγοντας άπληστα από την πολιτική $\pi\theta(a_i|s_i)$.

5.1.4. Προσέγγιση 4

Learning 2-opt Heuristics for the Traveling Salesman Problem via Deep Reinforcement Learning (da Costa et al., 2020)

Σε αυτήν την εργασία, προτείνεται ένας αλγόριθμος ενισχυμένης μάθησης που εκπαιδεύεται μέσω του Policy Gradient για την εκμάθηση βελτιστοποιημένων ευρετικών βασισμένων σε κινήσεις 2-opt. Η αρχιτεκτονική τους βασίζεται σε έναν μηχανισμό προσοχής δείκτη (Vinyals et al., 2015) που εξάγει τους κόμβους διαδοχικά για επιλογή ενεργειών. Εισάγουν μια μέθοδο ενισχυμένης μάθησης για να μάθουν μια στοχαστική πολιτική των επόμενων πολλά υποσχόμενων λύσεων, ενσωματώνοντας τις πληροφορίες ιστορικού αναζήτησης ενώ παρακολουθούν την τρέχουσα λύση με τις καλύτερες επισκέψεις. Τα αποτελέσματά τους δείχνουν ότι μπορούμε να μάθουμε πολιτικές για το Euclidean TSP που επιτυγχάνουν σχεδόν βέλτιστες λύσεις ακόμα και όταν ξεκινάμε από λύσεις κακής ποιότητας. Επιπλέον, η προσέγγισή τους μπορεί να επιτύχει καλύτερα αποτελέσματα από προηγούμενες μεθόδους βαθιάς μάθησης (Vinyals et al., 2015; Joshi et al., 2019; Kool et al., 2019; Deudon et al., 2018; Khalil et al., 2017, Bello and Pham, 2017). Η μέθοδός τους μπορεί εύκολα να προσαρμοστεί στη γενική k-opt και είναι πιο αποτελεσματική ως προς το δείγμα. Επιπλέον, οι πολιτικές που εκπαιδεύονται σε μικρά στιγμιότυπα μπορούν να επαναχρησιμοποιηθούν σε μεγαλύτερα στιγμιότυπα του TSP. Τέλος, η μέθοδός τους υπερέχει άλλων αποτελεσματικών ευρετικών, όπως τα OR-Tools της Google και είναι πολύ κοντά στις βέλτιστες λύσεις.

Οι συγγραφείς του παρόντος άρθρου, κωδικοποιούν πληροφορίες ακμών χρησιμοποιώντας συνελίξεις γραφημάτων και χρησιμοποιούν κλασική κωδικοποίηση ακολουθίας για να μάθουν τις αναπαραστάσεις των περιηγήσεων. Αποκωδικοποιούν αυτές τις αναπαραστάσεις μέσω ενός μηχανισμού προσοχής για να μάθουν μια στοχαστική πολιτική. Η προσέγγισή τους μοιάζει με την κλασική ευρετική 2-opt και μπορεί να ξεπεράσει τις προηγούμενες μεθόδους βαθιάς μάθησης σε ποιότητα λύσης και αποτελεσματικότητα δείγματος.

Τα Metaheuristics εξακολουθούν να απαιτούν ειδικές γνώσεις και μπορεί να έχουν υπο-βέλτιστους κανόνες στο σχεδιασμό τους. Για την αντιμετώπιση αυτού του περιορισμού, προτείνουν να συνδυαστεί η μηχανική μάθηση και οι 2-opt για την

εκμάθηση μιας στοχαστικής πολιτικής που αφορά τη διαδοχική βελτίωση των λύσεων TSP. Μια στοχαστική πολιτική μοιάζει με μια μετα-ευρετική όπου αποφεύγει δυνητικά τα τοπικά ελάχιστα. Η πολιτική τους επαναλαμβάνεται σε εφικτές λύσεις και η λύση ελάχιστου κόστους επιστρέφεται στο τέλος. Η κύρια ιδέα της μεθόδου τους είναι ότι λαμβάνοντας υπόψη μελλοντικές βελτιώσεις μπορεί να οδηγήσει σε καλύτερες πολιτικές από τις άπληστες ευρετικές.

Policy Gradient Neural Architecture

Το νευρωνικό τους δίκτυο, που βασίζεται σε μια αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή. Δύο μονάδες κωδικοποιητή χαρτογραφούν κάθε στοιχείο του $S = (S,S')$ ανεξάρτητα. Κάθε μονάδα διαβάζει τις εισόδους $X = (x_1,...,x_n)$, όπου x_i είναι συντεταγμένες κόμβων του κόμβου s_i στα S και S' . Στη συνέχεια, ο κωδικοποιητής μαθαίνει αναπαραστάσεις που ενσωματώνουν τόσο την τοπολογία του γραφήματος όσο και τη διάταξη κόμβων. Λαμβάνοντας υπόψη αυτές τις αναπαραστάσεις, ο αποκωδικοποιητής πολιτικής λαμβάνει δείγματα δεικτών ενεργειών $a_1, \ldots, a_k$ διαδοχικά, όπου $k = 2$ για 2-opt. Ο αποκωδικοποιητής τιμών λειτουργεί στις ίδιες εξόδους κωδικοποιητή αλλά εξάγει εκτιμήσεις πραγματικών τιμών.

Encoder

Ο σκοπός του κωδικοποιητή είναι να αποκτήσει μια αναπαράσταση για κάθε κόμβο στο γράφημα εισόδου δεδομένης της τοπολογικής του δομής και της θέσης του σε μια δεδομένη λύση. Για να επιτευχθεί αυτός ο στόχος, ενσωματώνουν στοιχεία από τα από τα συνελικτικά δίκτυα γραφημάτων (GCN) και την ενσωμάτωση ακολουθίας μέσω αναδρομικών νευρωνικών δικτύων (RNN).

Embedding Layers-Επίπεδο ενσωμάτωσης: Εισάγουν δισδιάστατες συντεταγμένες $x_i \in [0, 1]^2$, $\forall i \in 1, \ldots, n$, τα οποία είναι ενσωματωμένα σε χαρακτηριστικά d διαστάσεων ως $x^0 i = W_x x_i + b_x$, όπου $W_x \in R^{d\times 2}$, $b_x \in R^d$. Χρησιμοποιούν ως είσοδο τις Ευκλείδειες αποστάσεις $e_{i,j}$ μεταξύ των συντεταγμένων x_i και x_j για να προσθέσουν πληροφορίες ακμών και να ζυγίσουν τον πίνακα χαρακτηριστικών του κόμβου.

Graph Convolutional Layers-Συνελικτικά επίπεδα γραφήματος: Στα επίπεδα GCN, δηλώνουν ως x^l_i το διάνυσμα χαρακτηριστικών του κόμβου στο επίπεδο GCN l που σχετίζεται με τον κόμβο i. Στην είσοδο σε αυτά τα επίπεδα, έχουμε l = 0 και μετά τα επίπεδα L φτάνουμε σε παραστάσεις $z_i = x^L_i$ αξιοποιώντας τα χαρακτηριστικά κόμβου με την αναπαράσταση πρόσθετων χαρακτηριστικών ακμών.

Sequence Embedding Layers

Στη συνέχεια, χρησιμοποιούν ενσωματώσεις κόμβων z_i για να μάθουν μια αναπαράσταση ακολουθίας της εισόδου και να κωδικοποιήσουν μια περιήγηση. Λόγω συμμετρίας, μια περιήγηση από κόμβους (1, ... , n) έχει το ίδιο κόστος με την περιήγηση (n, ... , 1). Επομένως, διαβάζουν την ακολουθία και με τις δύο σειρές για να κωδικοποιήσουν ρητά τη συμμετρία μιας λύσης. Για να επιτευχθεί αυτός ο στόχος, χρησιμοποιούν δύο Long Short-Term Memory (LSTM), που υπολογίζονται χρησιμοποιώντας κρυφά διανύσματα από τον προηγούμενο κόμβο στην περιήγηση και την τρέχουσα ενσωμάτωση κόμβου.

Dual Encoder

Στη μεθοδολογία τους, μια κατάσταση $\bar{S}$ = (S, S’) αναπαρίσταται ως πλειάδα της τρέχουσας λύσης S και η καλύτερη λύση που έχει δει μέχρι στιγμής S’. Για το λόγο αυτό, κωδικοποιούν τόσο S όσο και S’ χρησιμοποιώντας ανεξάρτητα επίπεδα κωδικοποίησης. Έτσι, ορίζουν μια διαδοχική αναπαράσταση του S’ αφού περάσουν από επίπεδα κωδικοποίησης ως $h'_n \in R^d$.

Policy Decoder

Στοχεύουν στο να μάθουν τις παραμέτρους μιας στοχαστικής πολιτικής $\pi_\theta(A|\bar{S})$ που δίνοντας μια κατάσταση $\bar{S}$, εκχωρεί υψηλές πιθανότητες σε κινήσεις που μειώνουν το κόστος μιας περιήγησης. Ακολουθώντας τους Bello et al., η αρχιτεκτονική τους χρησιμοποιεί τον κανόνα της αλυσίδας για να παραγοντοποιήσει την πιθανότητα μιας κίνησης k-opt.

Pointing Mechanism

Χρησιμοποιούν έναν μηχανισμό κατάδειξης για να προβλέψουν μια κατανομή στις εξόδους κόμβων με κωδικοποιημένες ενέργειες (κόμβους) και μια αναπαράσταση κατάστασης.

Value Decoder

Παρόμοια με τον αποκωδικοποιητή πολιτικής, ο αποκωδικοποιητής αξίας τους λειτουργεί διαβάζοντας αναπαραστάσεις περιήγησης από S και S' και αναπαράσταση γραφήματος από το S. Δεδομένων των ενσωματώσεων Z, ο αποκωδικοποιητής τιμών λειτουργεί διαβάζοντας τις εξόδους z_i για κάθε κόμβο στην περιήγηση και τα κρυφά διανύσματα ακολουθίας h_n ,h'_n για να υπολογίσουν την αξία μιας κατάστασης. Χρησιμοποιούν μια μέση πράξη συγκέντρωσης για να συνδυάσουν τις αναπαραστάσεις κόμβων z_i σε μια αναπαράσταση γραφήματος. Αυτό το διάνυσμα στη συνέχεια συνδυάζεται με την αναπαράσταση περιήγησης h_v για να εκτιμηθούν οι τρέχουσες τιμές κατάστασης. Το δίκτυο αξίας εκπαιδεύεται σε έναν στόχο μέσω του Μέσου Τετραγωνικού Σφάλματος μεταξύ των προβλέψεών του.

5.2. Σύγκριση Μεθόδων

Στα προηγούμενα κεφάλαια αναλύθηκαν εκτενώς οι 4 προσεγγίσεις που επιλέχθηκαν από τη βιβλιογραφία που μελετήθηκε στα πλαίσια υλοποίησης της παρούσας εργασίας. Το παρόν κεφάλαιο θα σχολιάζει συνοπτικά τις βασικές διαφορές των 4 αυτών προσεγγίσεων που θα αποτελέσουν το υπόβαθρο των δικών μας υλοποιήσεων.

Αρχικά, το πρόβλημα προς επίλυση ήταν αυτό του TSP και το μοντέλο που υιοθετήθηκε ήταν αυτό του Pointer Network (Δίκτυο Δεικτών) για την προσέγγιση των Bello et al., (2017), Deudon et al. (2018) και da Costa et al. (2020), με την προσθήκη του Value Network όσον αφορά την ερευνητική δουλειά των da Costa et al. Η προσέγγιση με την Ιεραρχική Ενισχυμένη Μάθηση υιοθέτησε την τεχνική των Graph Pointer Network που αποτελεί εξέλιξη του Pointer Network. Σε κάθε πλαίσιο παρατηρούμε ότι ως είσοδος στο εκάστοτε μοντέλο είναι ένα τυχαίο σύνολο συντεταγμένων των εκάστοτε πόλεων που συμμετέχουν στην εκάστοτε διαδρομή. Ως έξοδο, έχουμε σε κάθε περίπτωση μία κατανομή σε μεταθέσεις διαφορετικών πόλεων. Πιο συγκεκριμένα,

προκύπτει είτε η επόμενη υποψήφια πόλη που προβλέπει το μοντέλο είτε η ακολουθία πόλεων που θεωρείται η πιο ωφέλιμη.

Τα νευρωνικά δίκτυα που σχεδιάστηκαν για να προβλέψουν τις παραμέτρους του RL μοντέλου σε κάθε paper, ήταν όλα LSTM με εξαίρεση την υλοποίηση των Deudon et al., που υιοθέτησαν έναν δικό τους μηχανισμό παραπλήσιο των LSTM. Στο άρθρο των da Costa et al., συναντήσαμε και τα Graph Convolutional Networks [4].

Σχεδόν σε όλες τις μεθόδους συναντήσαμε το δίκτυο προσοχής και τους κωδικοποιητές-αποκωδικοποιητές. Τα δύο πιο πρόσφατα papers χρησιμοποιούν ως υπόβαθρο το attention network μαζί με τους αντίστοιχους encoders-decoders και το εμπλουτίζουν αντίστοιχα με Graph Embedding Layers, Sequence Embedding Layers και Vector Context.

Η βελτιστοποίηση των παραμέτρων του Νευρωνικού Δικτύου που υιοθετήθηκε ανά περίπτωση, πραγματοποιήθηκε με τη Μέθοδο Policy Gradient και Ιεραρχικό Policy Gradient (3η Προσέγγιση) κάνοντας χρήση του αλγόριθμου Actor-Critic. Πιο συγκεκριμένα, η 1η προσέγγιση υιοθέτησε μία free policy based μέθοδο, η 2η τη Stochastic Gradient Descent μέθοδο, η 3η μία ιεραρχική policy based μέθοδο και η 4η ένα συνδυασμό της Policy Gradient μεθόδου με της 2-opt μεθόδου.

Οι στρατηγικές ανίχνευσης διαδρομών ποικίλλουν και συναντήσαμε:

- Για την 1η προσέγγιση τη μέθοδο της Δειγματοληψίας και της Ενεργής Αναζήτησης
- Για τη 2η προσέγγιση τη μέθοδο 2-opt Τοπικής Αναζήτησης και της Άπληστης μεθόδου
- Για τη 3η προσέγγιση τη μέθοδο της Δειγματοληψίας και της Άπληστης μεθόδου
- Για τη 4η προσέγγιση τη μέθοδο της Δειγματοληψίας

Οι 2 πρώτες προσεγγίσεις έκαναν χρήση της βιβλιοθήκης “Tensorflow”[5] και οι 2 τελευταίες της βιβλιοθήκης “Pytorch”. Η βιβλιοθήκη Pytorch παρατηρείται ότι έχει

[4] https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b
[5] https://www.tensorflow.org/

αποκτήσει μεγάλο αντίκτυπο τα τελευταία χρόνια και τείνει να αντικαταστήσει τη βιβλιοθήκη “Tensorflow”.

Καταλήγοντας, αξίζει να σημειωθεί ότι ο κώδικας της εκάστοτε προσέγγισης δινόταν ελεύθερα στο github.com και ότι όλες οι προσεγγίσεις χρειαζόντουσαν την ισχύ μίας κάρτας γραφικών ώστε να τρέξουν το μέγεθος των προβλημάτων TSP που περιέγραφαν. Μόνο το paper των Deudon et al., μπορούσε να τρέξει και σε περιβάλλον CPU.

RL PAPERS COMPARISON CHART

Features	NEURAL CO WITH RL (BELLO, 2017)	LEARNING HEURISTICS BY POLICY GRADIENT (DEUDON,2018)	GPNs AND HRL (COLUMBIA,2019)	LEARNING 2-OPT HEURISTICS VIA DRL (da COSTA,2020)
Problem	TSP	TSP	TSP	TSP
Model	Pointer Network	Pointer Network	Graph Pointer Network	Pointer Network, Value Network
Neural Network	RNN(LSTM cells)	Custom Mechanism	LSTM	LSTM
Input	Set of city coordinates	Set of city coordinates	Current city coordinates, all city coordinates	A sequence of n locations in a two dimensional space
Output(Prediction)	A distribution over different city permutations	A distribution over different city permutations	Probability distribution over the next candidate city	A permutation of the nodes
Reward	Minimizing tour length (negative reward)	Minimizing tour length (negative reward)	Different task(reward) for each level	Maximizing the best found solution(MDP)
Extra Model Technique	Attention(Encoder-Decoder)	Multi Head Attention(Encoder-Decoder), Feed Forward, Principal Component Analysis	Attention(Encoder-Decoder), Graph Embedding Layers, Vector Context	Encoder, Graph Convolutional Networks, Embedding Layers, Sequence Embedding Layers, Dual Encoding, Policy Decoder, Value Decoder
Optimizing parameters of the NN	Policy Gradient Method (Actor Critic Algorithm)	Policy Gradient Method. (Actor Critic Algorithm)	Hierarchical Policy Gradient Method (Layer-wise Policy Optimization)	Policy Gradient to learn improvement heuristics based on 2-opt moves
Training with RL model	Free Policy Based RL	Policy Based RL (SGD)	Hierarchical RL (different policy for each level)	Stochastic Policy (Solving the TSP via 2-opt as a Markov Decision Process (MDP))
Search Strategies	Sampling, Active Search	2-opt Local Search, Greedy	Sampling, Greedy	Sampling
Code	√	√	√	√
Library	Tensorflow	Tensorflow	Pytorch	Pytorch
GPU	√	√ (+CPU)	√	√

Εικόνα 7: Πίνακας Σύγκρισης Μεθοδολογιών

ΚΕΦΑΛΑΙΟ 6

6. Περιγραφή Υλοποίησης των Μεθόδων της βιβλιογραφίας και παρουσίαση νέας

6.1. Υλοποιήσεις βασισμένες στην βιβλιογραφία

Στο πρώτο υπο-κεφάλαιο του παρόντος κεφαλαίου, θα πραγματοποιηθεί μία εκτενής περιγραφή της 1ης μας επίλυσης του απλού TSP με Ενισχυμένη Μάθηση. Η επίλυση αυτή, βασίζεται στη μεθοδολογία των Bello et al., (2017) και Deudon et al., (2018). Στη συνέχεια, θα παρουσιαστεί μία υλοποίηση μας για το TSP με χρονικά παράθυρα που θα βασίζεται στη μέθοδο της Ιεραρχικής Ενισχυμένης Μάθησης. Τέλος, παρουσιάζεται μία επίλυση του προβλήματος του Περιοδεύοντος Πωλητή με LSTM νευρωνικά δίκτυα. Σε κάθε υπο-κεφάλαιο, συγκρίνονται τα αποτελέσματά μας με γνώστες μεθόδους επίλυσης προβλημάτων CO, όπως το OR-Tools.

6.1.1. Περιγραφή Επίλυσης TSP με Ενισχυμένη Μάθηση

Η πρώτη μας υλοποίηση βασίζεται στους Bello et al., (2017) και συγκεκριμένα στην αρχιτεκτονική των Deudon et al., (2018).

Στόχος μας είναι να μάθουμε τις παραμέτρους θ μιας στοχαστικής πολιτικής πάνω στις μεταθέσεις πόλεων $p_\theta(\pi|s)$, χρησιμοποιώντας Νευρωνικά Δίκτυα και Policy Gradient μέθοδο. Δίνεται ένα σύνολο σημείων εισαγωγής s όπου καλούμαστε να εκχωρήσουμε μεγαλύτερη πιθανότητα στις «καλές» περιηγήσεις π+ και μικρότερη πιθανότητα στις «ανεπιθύμητες» περιηγήσεις π− . Ο κωδικοποιητής αντιστοιχίζει ένα σύνολο εισόδου $I = (i_1,...,i_n)$ σε ένα σύνολο συνεχών αναπαραστάσεων $Z= (z_1,...,z_n)$. Με δεδομένο το Z, ο αποκωδικοποιητής δημιουργεί μια ακολουθία εξόδου $O =(o_1,...,o_n)$ συμβόλων δηλαδή ένα στοιχείο κάθε φορά.

Σε κάθε βήμα το μοντέλο είναι αυτόματα παλινδρομικό, χρησιμοποιώντας τα σύμβολα που δημιουργήθηκαν προηγουμένως ως πρόσθετη είσοδο κατά τη δημιουργία του επόμενου.

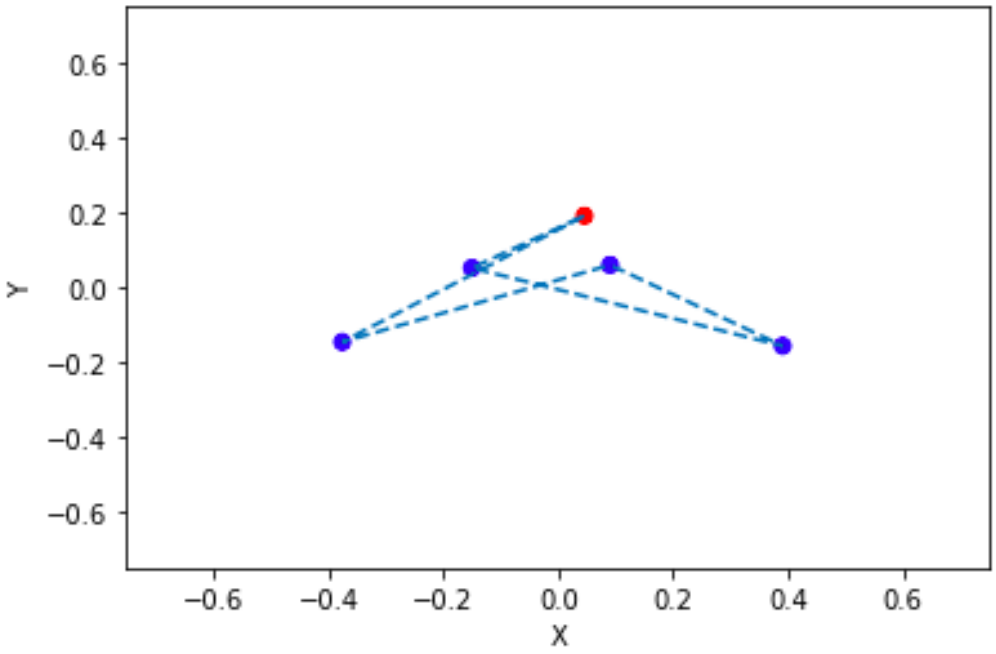
Παρακάτω απεικονίζεται με μεγαλύτερη λεπτομέρεια η αρχιτεκτονική που ακολουθήσαμε.

	Actor Embedding	**Actor Encoding**	**Encoder**	**Decoder**	**Pointer**	**Critic Embedding**	**Critic Encoding**
Input	Input Coordinates	Embedded Sequence	Actor Encoding	Encoded Reference	Decoder Output + Encoder Output	Input Coordinates	Embedded Sequence
Output	Embedded Sequence	Encoded Sequence	Reference for Actions	Encoded Query	Masked Scores (prediction of distribution)	Embedded Sequence	Encoding Sequence

Πίνακας 2: Περιγραφή του Μοντέλου μας

- Embedded input sequence [batch_size, seq_length, from_] -> [batch_size, seq_length, to_]
- Encode input sequence [batch_size, seq_length, n_hidden] -> [batch_size, seq_length, n_hidden]
- Encoder_output ->αποτελεί την αναφορά για τις δράσεις του actor [Batch size, Sequence Length, Num_neurons]
- Από μία ακολουθία (decoder output) [Batch size, n_hidden] και ένα σετ αναφορών (encoder_output) [Batch size, seq_length, n_hidden] -> προβλέπει μία κατανομή για το επόμενο decoder input.

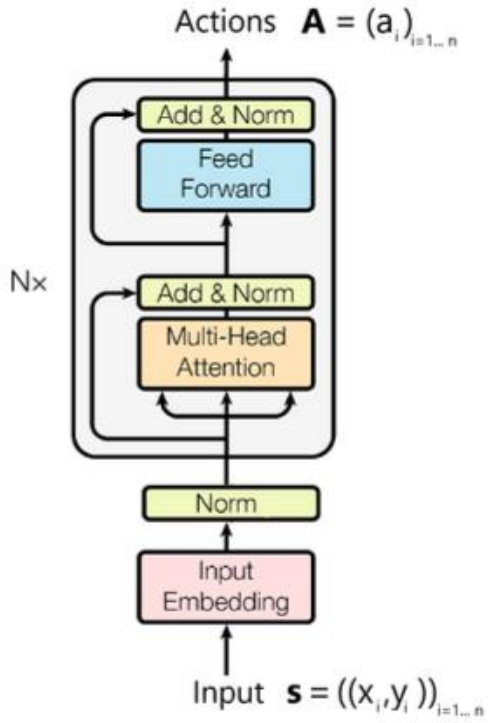
TSP SETTING AND INPUT PREPROCESSING



Εικόνα 8: Αναπαράσταση 5 πόλεων στο χώρο

Κάθε $city_i$ περιγράφεται από τις 2D συντεταγμένες του (x_i,y_i) σε έναν Ευκλείδειο χώρο. Για λόγους μνήμης και πόρων, θα δείξουμε ένα παράδειγμα 5 πόλεων.

Πρώτα, πρέπει να εκπαιδεύσουμε το μοντέλο μας και να κωδικοποιήσουμε τα δεδομένα μας. Έτσι, θα εξετάσουμε τον TSP encoder που προτάθηκε για πρώτη φορά από τους Deudon et al., (2018).



Εικόνα 9: TSP encoder

Ο κωδικοποιητής λαμβάνει μια αναπαράσταση για κάθε ενέργεια (πόλη) δεδομένου του πλαισίου της. Η έξοδος του κωδικοποιητή είναι ένα σύνολο διανυσμάτων δράσης A = (a_1,...,a_n), το καθένα αντιπροσωπεύει μια πόλη που αλληλοεπιδρά με άλλες πόλεις. Πιο συγκεκριμένα, κάθε είσοδος είναι ένα ενσωματωμένο και ομαλοποιημένο σύνολο n πόλεων.

Κάθε στρώμα έχει δύο υποστρώματα:

- 1ο υπόστρωμα->Multi-head Attention
- 2ο υπόστρωμα- >Τροφοδοσία-Προώθηση -> αποτελείται από δύο γραμμικούς μετασχηματισμούς κατά τη θέση, με ενεργοποίηση ReLU ενδιάμεσα.
- Η έξοδος κάθε υποστρώματος είναι LayerNorm(x + Sublayer(x)), όπου το Sublayer(x) είναι η συνάρτηση που υλοποιείται από το ίδιο το υπόστρωμα και το LayerNorm() σημαίνει κανονικοποίηση επιπέδου.

Attention and Multi-Head Attention

Το LSTM Attention, όπως φαίνεται από τη βιβλιογραφία (Bello et al., (2017)) που αναλύσαμε, επεξεργάζεται μια εισαγωγή τη φορά. Η προσοχή επιτρέπει την επεξεργασία βήμα προς βήμα ορισμένων περιοχών ή χαρακτηριστικών της εισόδου για τη λήψη πληροφοριών όταν και όπου χρειάζεται.

Σε κάθε βήμα, η επόμενη τοποθεσία επιλέγεται με βάση προηγούμενες πληροφορίες και απαιτήσεις για την εργασία.

Είναι γνωστό ότι η επεξεργασία της προσοχής στα lstm λαμβάνει υπόψιν όλες τις περιοχές και τα χαρακτηριστικά της εισόδου μία φορά, αλλά η προσοχή πολλαπλών κεφαλών παίρνει μια μέση εμφάνιση.

Οι μηχανισμοί νευρωνικής προσοχής επιτρέπουν στα ερωτήματα να αλληλοεπιδρούν με ζεύγη κλειδιών-τιμών. Για το TSP, τα ερωτήματα και τα ζεύγη κλειδιού-τιμής q_i , k_i , v_i $\in R^d$ λαμβάνονται μετατρέποντας γραμμικά κάθε $city_i \in R^d$ και εφαρμόζοντας μια μη γραμμικότητα ReLu.

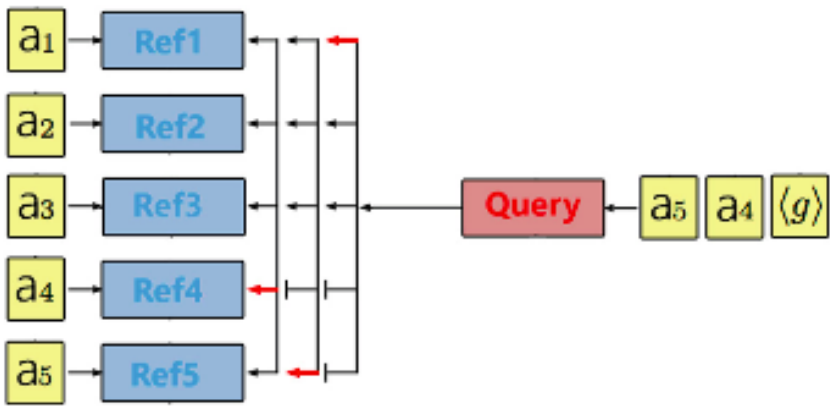
$$Attention(Q;K;V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

Decoder

Η αρχιτεκτονική μας χρησιμοποιεί τον κανόνα της αλυσίδας, όπως και οι Bello et al., (2017), για να παραγοντοποιήσει την πιθανότητα μιας περιήγησης ως:

$$p_\theta(\pi|s) = \prod_{t=1}^{n} p_\theta(\pi(t)|\pi(<t), s)$$

Κάθε όρος στη δεξιά πλευρά υπολογίζεται διαδοχικά με μονάδες softmax. Σε αντίθεση με τον αποκωδικοποιητή αρχιτεκτονικής των Bello et al., (2017) που συνοψίζει όλες τις προηγούμενες ενέργειες σε ένα διάνυσμα σταθερού μήκους, αυτό το μοντέλο ξεχνά μετά από K = 3 βήματα .



Εικόνα 10: TSP decoder

Σε κάθε χρόνο εξόδου t το μοντέλο αντιστοιχίζει τις 3 τελευταίες δειγματοληπτικές ενέργειες (επισκέψεις σε πόλεις) στο ακόλουθο διάνυσμα:

$$q_t = ReLu(W_1 a_{\pi(t-1)} + W_2 a_{\pi(t-2)} + W_3 a_{\pi(t-3)}) \in \mathbb{R}^{d'}$$

Pointing Mechanism

Ο μηχανισμός κατάδειξης χρησιμοποιείται ως νευρωνική αρχιτεκτονική για την εκμάθηση της υπό όρους πιθανότητας μιας ακολουθίας εξόδου με στοιχεία που είναι διακριτά και αντιστοιχούν σε θέσεις σε μια ακολουθία εισόδου.

Όπως αναφέρθηκε παραπάνω, το νευρωνικό δίκτυο περιλαμβάνει έναν κωδικοποιητή-αποκωδικοποιητή RNN συνδεδεμένο με μεγάλη προσοχή. Σε κάθε βήμα αποκωδικοποίησης, ένας «δείκτης» χρησιμοποιείται για δειγματοληψία από τον χώρο δράσης (στην περίπτωσή μας, μια κατανομή πιθανότητας στις πόλεις που πρέπει να επισκεφτούμε). Συνολικά παραμετροποιεί μια στοχαστική πολιτική για τις μεταθέσεις πόλεων $p_\theta(\pi|s)$. Η εκπαίδευση πραγματοποιείται από το Policy Gradient.

Ένα διάνυσμα q_t αλληλοεπιδρά με ένα σύνολο n διανυσμάτων για να ορίσει μια κατανομή κατάδειξης στον χώρο δράσης.

Μόλις γίνει δειγματοληψία της επόμενης πόλης, η τροχιά q_{t+1} ενημερώνεται με το επιλεγμένο διάνυσμα δράσης και η διαδικασία τελειώνει όταν ολοκληρωθεί η

περιήγηση. Ο στόχος είναι να προβλέψουμε μια κατανομή στο σύνολο των n διανυσμάτων ενεργειών, με δεδομένο ένα διάνυσμα q_t.

Για το παρακάτω τύπο, ας λάβουμε υπόψη ότι το u_t είναι το διάνυσμα προσοχής που υπολογίστηκε από τη διαδικασία προσοχής πολλαπλών κεφαλών.

$$p_\theta(\pi(t)|\pi(<t), s) = softmax(C\,tanh(u^t/T))$$

Training Procedure

Επιλέγουμε να εκπαιδεύσουμε το NN μας με τη μέθοδο Policy Gradient χρησιμοποιώντας τον κανόνα μάθησης REINFORCE με τη βοήθεια ενός critic ώστε να μειώσουμε τη διακύμανση των διαβαθμίσεων. Για το TSP, χρησιμοποιούμε τη διάρκεια της περιήγησης ως ανταμοιβή r(π|s) = L(π|s) , την οποία επιδιώκουμε να ελαχιστοποιήσουμε.

Ο στόχος της εκπαίδευσής μας είναι η αναμενόμενη ανταμοιβή που με βάση ένα γράφημα εισαγωγής ορίζεται ως:

$$J(\theta|s) = \mathbb{E}_{\pi\sim p_\theta(.|s)}[r(\pi|s)]$$

Actor-Critic

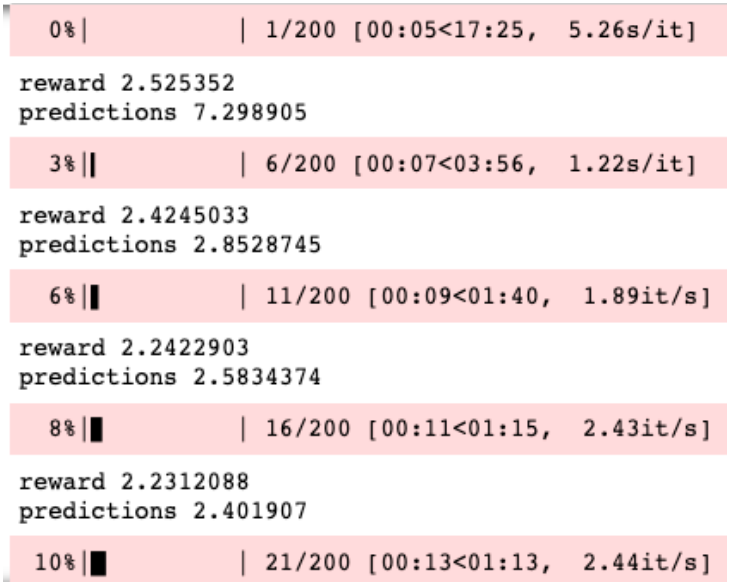
Actor: Μαθαίνουμε τις παραμέτρους του actor θ ξεκινώντας από μια τυχαία πολιτική και βελτιστοποιώντας τες επαναληπτικά με τον κανόνα μάθησης REINFORCE και Stochastic Gradient Descent (SGD), σε περιπτώσεις που δημιουργούνται εν κινήσει. Στη συνέχεια εφαρμόζουμε backpropagation(παραπομπή) στις ενέργειες.

Critic: Ο critic χρησιμοποιεί τον ίδιο κωδικοποιητή με τον actor. Ο critic εκπαιδεύεται ελαχιστοποιώντας το Μέσο Τετραγωνικό Σφάλμα μεταξύ των προβλέψεών του και των ανταμοιβών του actor. Στην περίπτωση του critic εφαρμόζουμε backpropagation στις ανταμοιβές. Ο critic εκτιμά την απόδοση του actor.

Results-Training & Testing

Training:

1. Η επιβράβευση μας είναι το μήκος της διαδρομής.
2. Η πρόβλεψή μας είναι η εκτίμηση της επίδοσης του μοντέλου μας.

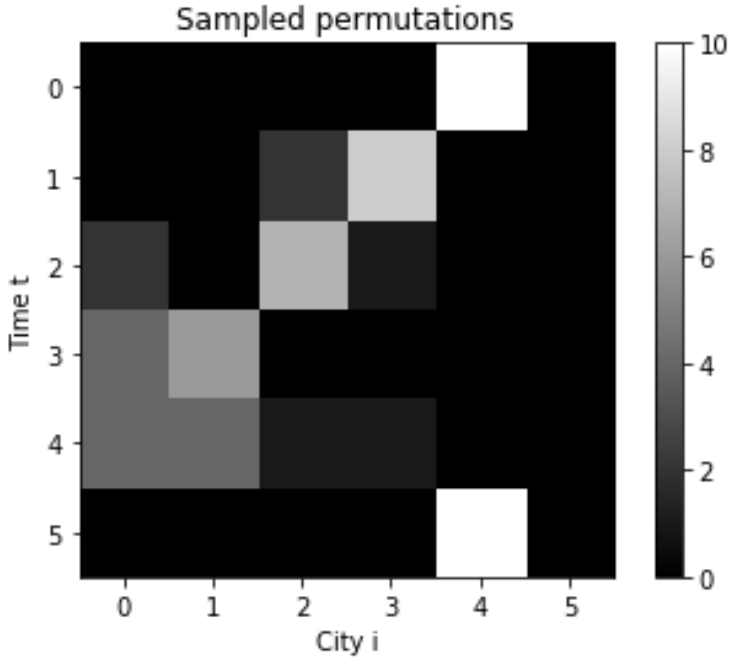


Εικόνα 11: Παράδειγμα Εκπαίδευσης στο Περιβάλλον του Jupyter Notebook

Coordinates of the predicted solution and visualization

	1	2	3	4	5
x	0,30184938	0,35127783	-0,05210716	-0,2651541	-0,33586596
y	0,03602304	0,02694098	-0,07697405	-0,27778521	0,29179524

Εικόνα 12: Συντεταγμένες του παραδείγματός μας.

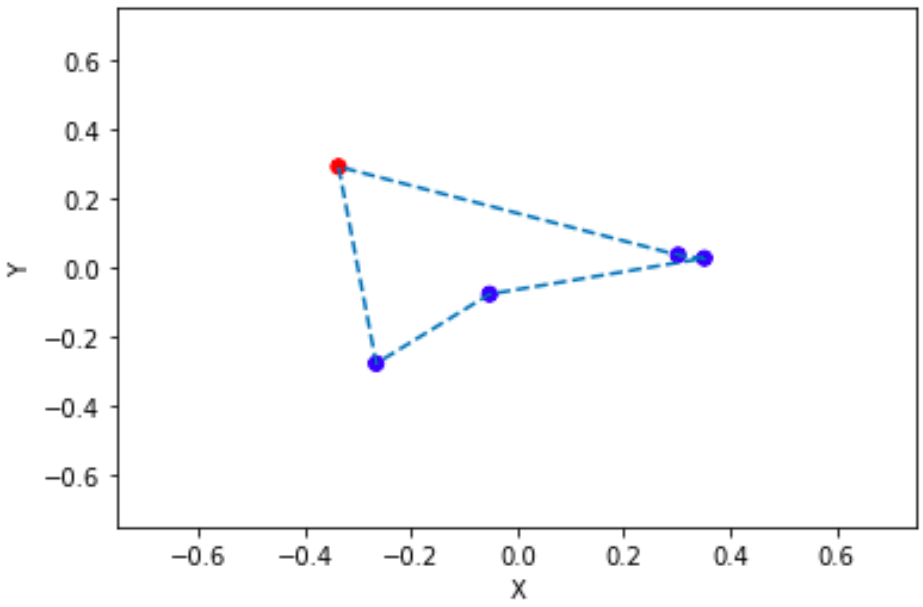


Εικόνα 13: Απεικόνιση Μονοπατιού

```
[4 3 2 1 0]
[[-0.33586596  0.29179524]
 [-0.2651541  -0.27778521]
 [-0.05210716 -0.07697405]
 [ 0.35127783  0.02694098]
 [ 0.30184938  0.03602304]]
```

Εικόνα 14: Σειρά Επίσκεψης Πόλεων βάσει Συντεταγμένων

Η επιβράβευση, η οποία είναι το μήκος της διαδρομής μας είναι: 2.02.



Εικόνα 15: Απεικόνιση Βέλτιστης Διαδρομής στο Χώρο

Προκειμένου να συγκρίνουμε την υλοποίησή μας, επιλέξαμε να λύσουμε το πρόβλημα TSP με τις ίδιες συντεταγμένες σε έναν Excel Solver.

Οι αποστάσεις μεταξύ των πόλεων είναι οι εξής:

	1	2	3	4	5
1	0	0,0502559	0,371555614	0,648049816	0,687095534
2	0,0502559	0	0,416554659	0,687638259	0,736419967
3	0,371555614	0,416554659	0	0,29276974	0,465306185
4	0,648049816	0,687638259	0,29276974	0	0,573953009
5	0,687095534	0,736419967	0,465306185	0,573953009	0

Πίνακας 3: Αποστάσεις των Πόλεων

Η βέλτιστη διαδρομή είναι (Optimal Tour):

4	3	1	2	5
0,29276974	0,371555614	0,0502559	0,736419967	0

Πίνακας 4: Βέλτιστη Διαδρομή

Το βέλτιστο μήκος διαδρομής είναι =1.45

Multi-Head Attention Pointer Net	Excel Solver
5 city-coordinates	5 city-coordinates
Reward-tour length: 2.02	**Optimal Tour Length: 1.45**

Πίνακας 5: Σύγκριση Μεθόδων

Μπορεί να φαίνεται ότι έχουμε ένα τεράστιο σφάλμα (39,3%). Ένα πολύ μικρό σύνολο δεδομένων μπορεί να επηρεάσει δραματικά την απόδοση ενός NN. Τα NN χρειάζονται μεγάλα και διάφορα σύνολα δεδομένων για να μάθουν και να αποδώσουν κοντά στις βέλτιστες ή ακριβείς λύσεις.

6.1.2. Περιγραφή Επίλυσης TSPTW (TSP with Time Windows) με Ενισχυμένη Μάθηση

Αρχικά θα εξηγηθεί η μορφή του TSPTW προβλήματος. Κάθε κόμβος i έχει το δικό του χρονικό διάστημα εξυπηρέτησης $[e_i , l_i]$, όπου e_i είναι ο χρόνος άφιξης και l_i ο χρόνος αποχώρησης. Μια πόλη δεν είναι επισκέψιμη μετά την ώρα της αναχώρησής πωλητή από αυτή.

Εάν ο κόμβος επισκεφθεί νωρίτερα από την ώρα άφιξης, ο πωλητής πρέπει να περιμένει μέχρι να ξεκινήσει η υπηρεσία, μέχρι την ώρα άφιξης.

Για δεδομένα TSPTW, κάθε ένας από τους κόμβους x_i είναι μια πλειάδα (x_i, y_i, e_i, l_i), όπου (x_i, y_i) είναι μια δισδιάστατη συντεταγμένη και e_i, l_i είναι ο χρόνος άφιξης και αποχώρησης.

Πιο συγκεκριμένα, έχουμε την παρακάτω δομή:

$$\begin{aligned} \min_{\sigma} \quad & \sum_{i=1}^{N} c_i \\ \text{s.t.} \quad & c_{i+1} - c_i \geq \|\mathbf{x}_{\sigma(i+1)} - \mathbf{x}_{\sigma(i)}\|_2, \quad i \in \{1, \ldots, N-1\}, \\ & e_i \leq c_i \leq l_i \qquad\qquad\qquad\quad i \in \{1, \ldots, N\}, \end{aligned}$$

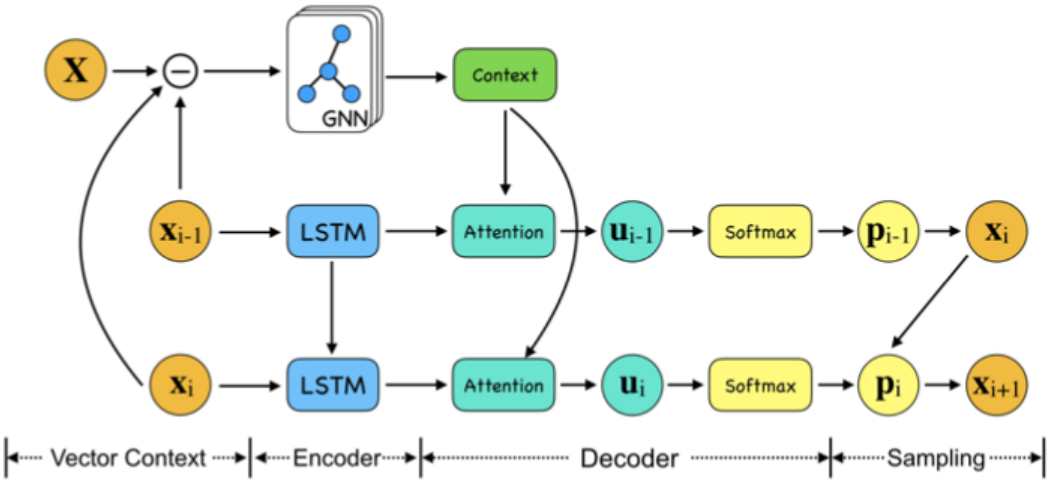
• όπου c_i είναι το κόστος χρόνου για την i-η πόλη.

• Μια εφικτή λύση δεν υπάρχει πάντα. Για να εξασφαλιστεί η ύπαρξη δεδομένων εκπαίδευσης και δοκιμής δημιουργούμε πρώτα στιγμιότυπα TSP20 από ομοιόμορφη κατανομή $[0, 1]^2$.

• Στη συνέχεια, χρησιμοποιώντας τοπική αναζήτηση 2-opt στα παραγόμενα στιγμιότυπα, λύνουμε τις κατά προσέγγιση λύσεις c_i για $i \in \{1, ..., N\}$.

• $e_i \leq c_i \leq l_i$, που σημαίνει ότι υπάρχουν πάντα εφικτές λύσεις στα δεδομένα εκπαίδευσης και δοκιμής.

Model Architecture and Training



Εικόνα 16: Αρχιτεκτονική GPN

- Η αρχιτεκτονική του GPN αποτελείται από :
  - Encoder
  - Graph Embedding Layer
  - Vector Context
  - Decoder
- Η τρέχουσα συντεταγμένη πόλης x_i κωδικοποιείται από το LSTM ενώ το X κωδικοποιείται ως το διανυσματικό πλαίσιο από ένα νευρωνικό δίκτυο γραφήματος.
- Τα κωδικοποιημένα διανύσματα περνούν στον αποκωδικοποιητή προσοχής, ο οποίος εξάγει το διάνυσμα δείκτη.

- Η κατανομή πιθανοτήτων στην επόμενη υποψήφια πόλη είναι $p_i = \text{softmax}(u_i)$.
- Η επόμενη πόλη που επισκεφθήκαμε x_{i+1} λαμβάνεται δειγματοληπτικά από το p_i.

Τα δεδομένα εκπαίδευσης παράγονται τυχαία από μια ομοιόμορφη κατανομή $[0, 1]^2$.

Σε κάθε εποχή, τα δεδομένα εκπαίδευσης παράγονται εν κινήσει. Για GPN ενός επιπέδου, η συνάρτηση ανταμοιβής περιλαμβάνει τόσο την ποινή όσο και τον στόχο του TSPTW. Πιο συγκεκριμένα, περιλαμβάνει την τιμωρία εάν ο χρόνος άφιξης υπερβαίνει τον χρόνο αναχώρησης + το συνολικό κόστος χρόνου όλου του TSPTW).

Comparison with OR-Tools

Για σύγκριση TSP με Time Windows (TSPTW), ο αλγόριθμος Savings επιλέγεται ως η πρώτη στρατηγική λύσης στο OR-Tools.

- Χρησιμοποιήσαμε την προεπιλεγμένη ρύθμιση για τα όρια αναζήτησης και τα μεταευρετικά.
- Παρατηρούμε ότι η επίλυση μας που αφορά την πρόβλεψη για τη σωστή επίλυση του TSP, είναι αρκετά κοντά στη βέλτιστη λύση του OR-Tools.

GPN	OR-Tools
Random Data Generator (same function, size, format)	Random Data Generator (same function, size, format)
Use 2-opt local search on the generated instances	Use 2-opt local search on the generated instances
Total time cost: 3.32	**(Best) Total time cost: 3.19**

Εικόνα 17: Σύγκριση GPN με OR-Tools

6.2. Περιγραφή Επίλυσης TSP με Νευρωνικά Δίκτυα

Η τελευταία υλοποίηση αφορά την επίλυση του TSP με LSTM[6] νευρωνικά δίκτυα. Ο κώδικας που παραμετροποιήσαμε ώστε να λειτουργήσει προς όφελός μας, προέρχεται από το github[7]. Θέσαμε τον αριθμό των κόμβων δηλαδή των πόλεων να είναι 10. Οι συντεταγμένες των πόλεων που εμπεριέχονται σε ξεχωριστό αρχείο, δίνονται ως είσοδο στο μοντέλο μας αφού πρώτα υπολογιστούν οι αποστάσεις μεταξύ τους. Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι η tensorlow[8] και η sklearn.metrics[9].

Η αρχιτεκτονική του μοντέλου μας ανά επίπεδο φαίνεται ξεκάθαρα στο παρακάτω σχήμα. Ουσιαστικά παρατηρούμε ότι αποτελείται από 9 στρώματα (layers). Όπως γνωρίζουμε η είσοδος για τα LSTM πρέπει να είναι τρισδιάστατη. Για αυτό σε περίπτωση εισόδου 2D, μεταβάλλεται σε 3D για να λειτουργήσει.

Η σειρά τους είναι ενδεικτική και επιλέχθηκε ύστερα από πειράματα. Παρακάτω παρουσιάζεται επιγραμματικά τι συμβαίνει σε κάθε επίπεδο.

- inputs = keras.Input(shape=(10, 10,)) -> Input[10] των δεδομένων μας και μορφοποίηση μεγέθους.
- time_dist_1 = TimeDistributed(Dense(50, activation='relu'))(inputs) -> Το TimeDistributedDense[11] εφαρμόζει την ίδια λειτουργία με το Dense (πλήρως συνδεδεμένο) σε κάθε χρονικό βήμα ενός τρισδιάστατου tensor. Η χρήση του στρώματος περιτυλίγματος TimeDistributed αφορά την ανάγκη για ορισμένα επίπεδα LSTM να επιστρέφουν ακολουθίες αντί για μεμονωμένες τιμές.
- time_dist = Conv1D(32, 3, padding='same')(time_dist_1) -> Το στρώμα Conv1D[12] δημιουργεί έναν πυρήνα συνέλιξης που συνελίσσεται με την είσοδο του επιπέδου σε μια ενιαία χωρική (ή χρονική) διάσταση για να παράγει έναν tensor εξόδων. Εάν το use_bias είναι True, δημιουργείται ένα διάνυσμα πόλωσης και

[6] https://keras.io/api/layers/recurrent_layers/lstm/
[7] https://gist.github.com/mlalevic/6222750
[8] https://www.tensorflow.org/
[9] https://scikit-learn.org/stable/index.html
[10] https://keras.io/api/models/model/
[11] https://keras.io/api/layers/recurrent_layers/time_distributed/
[12] https://keras.io/api/layers/convolution_layers/convolution1d/

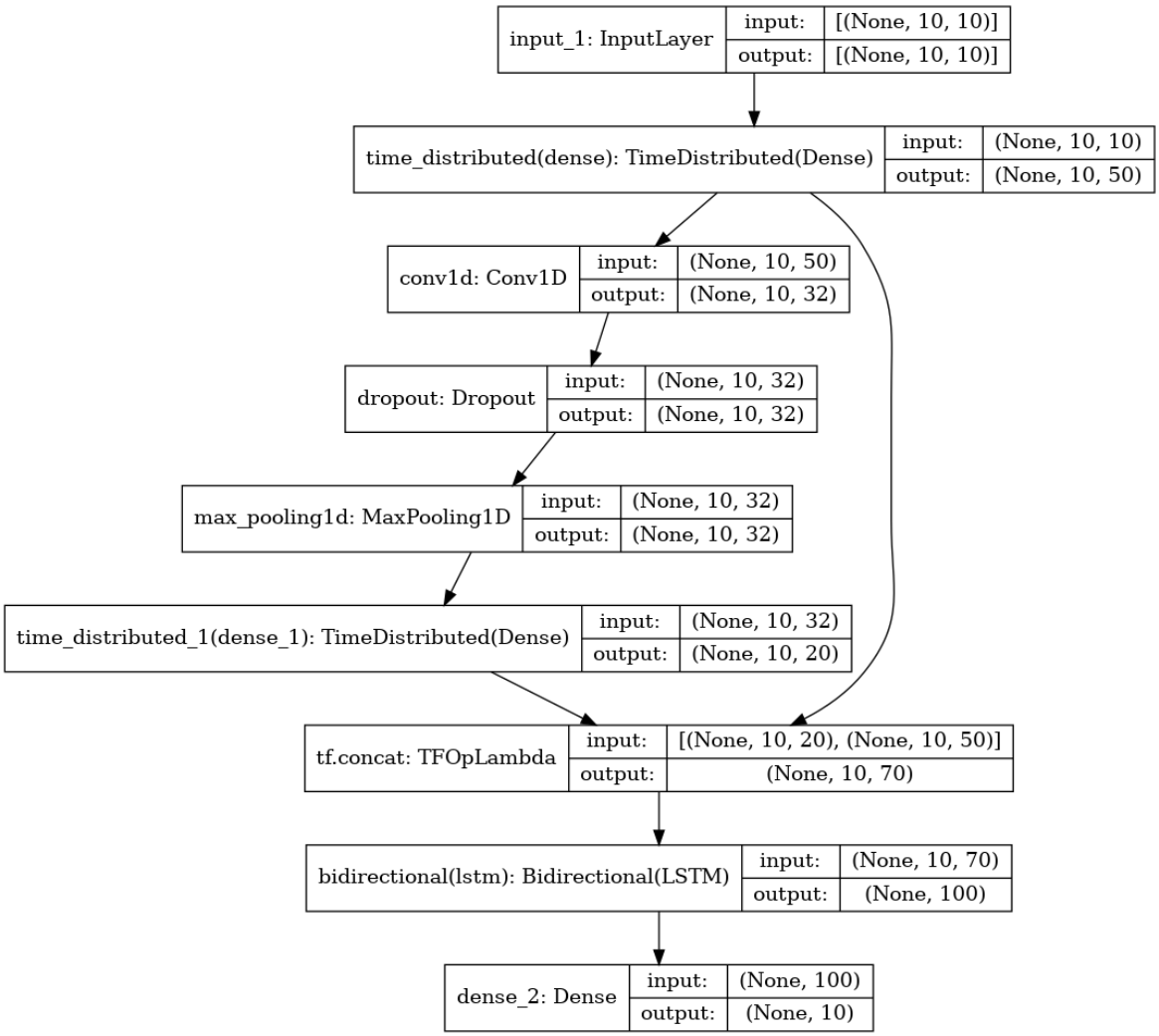
προστίθεται στις εξόδους. Τέλος, αν η ενεργοποίηση δεν είναι None, εφαρμόζεται και στις εξόδους μας.

- time_dist = Dropout(0.5)(time_dist) -> Εφαρμόζει το Dropout[13] στις τιμές εισόδου. Το Dropout ορίζει τυχαία τις μονάδες εισόδου στο 0 σε κάθε βήμα κατά τη διάρκεια του training, κάτι που βοηθά στην αποφυγή υπερβολικής προσαρμογής δηλαδή overfitting.
- time_dist = MaxPooling1D(pool_size=2,strides=1, padding='same')(time_dist)-> Το Max pooling[14] είναι μια διαδικασία διακριτοποίησης. Ο στόχος είναι «μικρύνει» το δείγμα μιας αναπαράστασης εισόδου. Έτσι μειώνεται η διάστασή της και επιτρέπεται να γίνουν υποθέσεις σχετικά με τα χαρακτηριστικά που περιέχονται στις υπο-περιοχές.
- time_dist = TimeDistributed(Dense(20, activation='relu'))(time_dist)-> Ίδιο με παραπάνω με την προσθήκη της επεξήγησης για τη ReLU[15] activation function. Η συνάρτηση ReLU εφαρμόζει τη λειτουργία ενεργοποίησης της μη γραμμικότητας. Με τις προεπιλεγμένες τιμές, αυτό επιστρέφει την τυπική ενεργοποίηση ReLU: max(x, 0), δηλαδή το μέγιστο στοιχείο ανάμεσα στο 0 και το tensor εισόδου μας. Η τροποποίηση των προεπιλεγμένων παραμέτρων μας επιτρέπει να χρησιμοποιούμε μη μηδενικά όρια, να αλλάζουμε τη μέγιστη τιμή της ενεργοποίησης και να χρησιμοποιούμε ένα μη μηδενικό πολλαπλάσιο της εισόδου για τιμές κάτω από το όριο που θέσαμε.
- time_dist = concatenate([time_dist, time_dist_1], axis=-1) -> Το concatenate[16] στρώμα συνενώνει μια λίστα εισόδων. Λαμβάνει ως είσοδο μια λίστα tensors, όλοι έχουν το ίδιο σχήμα εκτός από τον άξονα συνένωσης, και επιστρέφει έναν μόνο tensor που είναι η συνένωση όλων των εισόδων.
- decoder_outputs = Bidirectional(LSTM(50))(time_dist)-> Η χρήση αμφίδρομης λειτουργίας[17] θα τρέξει τις εισόδους μας με δύο τρόπους, έναν από το παρελθόν στο μέλλον και έναν από το μέλλον στο παρελθόν και σε αυτό που διαφέρει

[13] https://keras.io/api/layers/regularization_layers/dropout/
[14] https://keras.io/api/layers/pooling_layers/max_pooling1d/
[15] https://keras.io/api/layers/activation_layers/relu/
[16] https://keras.io/api/layers/merging_layers/concatenate/
[17] https://stackoverflow.com/questions/43035827/whats-the-difference-between-a-bidirectional-lstm-and-an-lstm

αυτήν η προσέγγιση από τη μονο-κατευθυντική είναι ότι στο LSTM που τρέχει προς τα πίσω διατηρεί πληροφορίες από το μέλλον.

- decoder_dense = Dense(num_decoder_tokens, activation='softmax')-> Απλώς το κανονικό μας στρώμα NN με πυκνή σύνδεση[18]. Το Dense υλοποιεί τη λειτουργία: output = activation (dot(input, kernel) + bias) όπου η ενεργοποίηση είναι η συνάρτηση ενεργοποίησης βάσει στοιχείων που μεταβιβάζεται ως όρισμα ενεργοποίησης. Ο πυρήνας είναι ένας πίνακας βαρών που δημιουργείται από το επίπεδο και το bias είναι ένα διάνυσμα πόλωσης που δημιουργείται από το επίπεδο (ισχύει μόνο εάν το use_bias είναι True). Όλα αυτά είναι χαρακτηριστικά του Dense.



Εικόνα 18:Model Plot

[18] https://keras.io/api/layers/core_layers/dense/

Αφού αναλύθηκε η αρχιτεκτονική του μοντέλου μας, παρουσιάζεται το αποτέλεσμα ύστερα από 10 epochs που έτρεξε το μοντέλο μας. Στο τελευταίο epoch διαφαίνεται ότι το μοντέλο στο validation set που αποτελεί το 20% των αρχικών δεδομένων που χρησιμοποιήσαμε, πετυχαίνει score 85%. Πιο συγκεκριμένα, το μοντέλο μας μπορεί και προβλέπει τη βέλτιστη διαδρομή του Περιοδεύοντος Πωλητή μας με ακρίβεια 85%. Γενικά, τα score από 85% και πάνω θεωρούνται αξιόλογα και λαμβάνονται υπόψη. Με μεγαλύτερο σύνολο δεδομένων, διαφορετικές παραμέτρους και χρήση κάρτας γραφικών, θα μπορούσαμε να επιτύχουμε ενδεχομένως και καλύτερα score.

```
Epoch 1/10
6250/6250 [==============================] - 105s 16ms/step - loss: 0.2762 - accuracy: 0.2638 - val_loss: 0.1661 - val_accuracy: 0.6334

Epoch 2/10
6250/6250 [==============================] - 99s 16ms/step - loss: 0.1507 - accuracy: 0.6733 - val_loss: 0.1179 - val_accuracy: 0.7572

Epoch 3/10
6250/6250 [==============================] - 100s 16ms/step - loss: 0.1084 - accuracy: 0.7793 - val_loss: 0.0923 - val_accuracy: 0.8120

Epoch 4/10
6250/6250 [==============================] - 99s 16ms/step - loss: 0.0913 - accuracy: 0.8123 - val_loss: 0.0849 - val_accuracy: 0.8250

Epoch 5/10
6250/6250 [==============================] - 99s 16ms/step - loss: 0.0828 - accuracy: 0.8290 - val_loss: 0.0793 - val_accuracy: 0.8345

Epoch 6/10
6250/6250 [==============================] - 100s 16ms/step - loss: 0.0778 - accuracy: 0.8386 - val_loss: 0.0780 - val_accuracy: 0.8349

Epoch 7/10
6250/6250 [==============================] - 100s 16ms/step - loss: 0.0740 - accuracy: 0.8466 - val_loss: 0.0734 - val_accuracy: 0.8452

Epoch 8/10
6250/6250 [==============================] - 100s 16ms/step - loss: 0.0717 - accuracy: 0.8499 - val_loss: 0.0719 - val_accuracy: 0.8478

Epoch 9/10
6250/6250 [==============================] - 102s 16ms/step - loss: 0.0700 - accuracy: 0.8530 - val_loss: 0.0702 - val_accuracy: 0.8525

Epoch 10/10
6250/6250 [==============================] - 101s 16ms/step - loss: 0.0687 - accuracy: 0.8559 - val_loss: 0.0706 - val_accuracy: 0.8513
```

Εικόνα 19: Ανάλυση Εποχών

ΚΕΦΑΛΑΙΟ 7

7.1. Συμπεράσματα

Παρατηρώντας τα αποτελέσματα των πειραμάτων που τρέξαμε στα πλαίσια υλοποίησης της παρούσας μεταπτυχιακής εργασίας, αντιλαμβανόμαστε ότι πλέον τα προβλήματα της Συνδυαστικής Βελτιστοποίησης μπορούν να επιλυθούν με αρκετά μεγάλη ακρίβεια μέσω μεθόδων Deep Learning. Πιο συγκεκριμένα, νευρωνικά δίκτυα και Ενισχυμένη Μάθηση αποτελούν τους πιο κατάλληλους υποψηφίους για την επίλυση τέτοιων προβλημάτων. Οι προβλέψεις για το βέλτιστο μονοπάτι που θα πρέπει να ακολουθήσει ο Πωλητής αγγίζουν αρκετά μεγάλη ακρίβεια που συναγωνίζεται την ακριβή επίλυση του προβλήματος μέσω κλασικών μεθόδων CO. Συμπερασματικά, θα μπορούσαμε να πούμε ότι οι μέθοδοι και οι προσεγγίσεις που περιγράψαμε παραπάνω θα αποτελέσουν το μέλλον για την επίλυση τέτοιων προβλημάτων.

Στο συγκεκριμένο σημείο αξίζει να σημειωθεί ότι η επίλυση τέτοιων προβλημάτων με τις μεθόδους και τις προσεγγίσεις που παρουσιάστηκαν παραπάνω, απαιτεί αρκετά μεγάλη υπολογιστική ισχύ και πόρους. Η χρήση κάρτας γραφικών (GPU) κρίνεται αναγκαία καθώς σε όλα τα ερευνητικά άρθρα που μελετήθηκαν, παρατηρείται η χρήση ενός και περισσότερων καρτών γραφικών ώστε να εκπαιδευτούν τα μοντέλα ορθώς. Τα στιγμιότυπα μεγάλης κλίμακας δεν μπόρεσαν να αναπαρασταθούν λόγω έλλειψης πόρων για αυτό και προτιμήσαμε τα προβλήματα μικρής κλίμακας. Αδιαμφισβήτητα, η εκπαίδευση σε μεγαλύτερη κλίμακα και με καλύτερους πόρους θα είχε αποφέρει μεγαλύτερη ακρίβεια.

Ερευνήθηκε και επισημάνθηκε πώς μπορεί να χρησιμοποιηθεί η μηχανική μάθηση και πιο συγκεκριμένα η Ενισχυμένη Μάθηση για τη δημιουργία συνδυαστικών αλγορίθμων βελτιστοποίησης που μαθαίνονται. Έχουμε υπογραμμίσει ότι η μάθηση μίμησης από μόνη της μπορεί να είναι πολύτιμη εάν η πολιτική που μαθαίνεται είναι σημαντικά ταχύτερη στον υπολογισμό από την αρχική που παρέχεται από έναν ειδικό. Σε αυτήν την περίπτωση εννοούμε έναν αλγόριθμο συνδυαστικής βελτιστοποίησης. Αντίθετα, τα μοντέλα που εκπαιδεύονται μέσω ανταμοιβής έχουν τη δυνατότητα να ξεπεράσουν τις τρέχουσες πολιτικές, δεδομένης της αρκετής εκπαίδευσης και μιας

εποπτευόμενης αρχικοποίησης. Η εκπαίδευση μιας πολιτικής που γενικεύεται σε αόρατα προβλήματα είναι μια πρόκληση, γι' αυτό πιστεύουμε ότι η μάθηση πρέπει να γίνεται σε μια κατανομή αρκετά μικρή ώστε η πολιτική να μπορεί να εκμεταλλευτεί πλήρως τη δομή του προβλήματος και να δώσει καλύτερα αποτελέσματα. Θεωρείται ότι οι προσεγγίσεις μηχανικής μάθησης για συνδυαστική βελτιστοποίηση μπορούν να βελτιωθούν χρησιμοποιώντας τη μηχανική μάθηση σε συνδυασμό με τους τρέχοντες αλγόριθμους συνδυαστικής βελτιστοποίησης ώστε να επωφεληθούν από τις θεωρητικές νόρμες και τους αλγόριθμους state-of-the-art που είναι ήδη διαθέσιμοι.

Οι πιο πρόσφατες εργασίες που χρησιμοποιούν τη βαθιά μάθηση για την επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή (TSP) έχουν επικεντρωθεί στην εκμάθηση της κατασκευής των ευρετικών. Τέτοιες προσεγγίσεις βρίσκουν λύσεις TSP καλής ποιότητας, αλλά απαιτούν πρόσθετες διαδικασίες, όπως «Beam Search» και «Sampling» για τη βελτίωση των λύσεων και την επίτευξη καλύτερης απόδοσης. Ωστόσο, λίγες μελέτες έχουν επικεντρωθεί στην βελτίωση των ευρετικών, όπου μια δεδομένη λύση βελτιώνεται μέχρι να φτάσει σε μια σχεδόν βέλτιστη.

Αν και οι περισσότερες από τις προσεγγίσεις που συζητήσαμε σε αυτή την εργασία εξακολουθούν να βρίσκονται σε διερευνητικό επίπεδο ανάπτυξης, τουλάχιστον όσον αφορά τη χρήση τους σε λύτες γενικής χρήσης (εμπορικούς), πιστεύουμε ακράδαντα ότι αυτή είναι μόνο η αρχή μιας νέας εποχής για τους αλγόριθμους Συνδυαστικής Βελτιστοποίησης.

Εν κατακλείδι, η Συνδυαστική Βελτιστοποίηση (CO) είναι το εργαλείο πολλών σημαντικών εφαρμογών στην επιχειρησιακή έρευνα, τη μηχανική και άλλους τομείς και, ως εκ τούτου, έχει προσελκύσει τεράστια προσοχή από την ερευνητική κοινότητα πρόσφατα. Ορισμένες αποτελεσματικές προσεγγίσεις σε κοινά προβλήματα περιλαμβάνουν τη χρήση ευρετικών για τη διαδοχική κατασκευή μιας λύσης. Ως εκ τούτου, είναι ενδιαφέρον να δούμε πώς ένα πρόβλημα CO μπορεί να αναδιατυπωθεί ως μια διαδοχική διαδικασία λήψης αποφάσεων και εάν αυτές οι ευρετικές μπορούν να μαθευτούν από έναν agent Ενισχυμένης Μάθησης (RL). Αυτή η εργασία διερεύνησε τη συνέργεια μεταξύ των πλαισίων CO και RL, η οποία μπορεί να γίνει μια πολλά υποσχόμενη κατεύθυνση για την επίλυση συνδυαστικών προβλημάτων.

7.2. Μελλοντική Έρευνα

Τα προηγούμενα κεφάλαια έχουν καλύψει διάφορες προσεγγίσεις για την επίλυση του προβλήματος του Περιοδεύοντος Πωλητή χρησιμοποιώντας αλγόριθμους Ενισχυμένης Μάθησης. Αυτό το πεδίο αναπτύσσεται ταχέως και αναμένουμε την εμφάνιση νέων αλγορίθμων και προσεγγίσεων για την αντιμετώπιση πολλών αδυναμιών και περιορισμών των τρεχουσών ερευνητικών εργασιών. Ένα από αυτά τα ζητήματα που έχει προκύψει, είναι ο χειρισμός μεγάλων στιγμιοτύπων του εκάστοτε προβλήματος από την άποψη του υπολογιστικού χρόνου, καθώς είναι ένας σημαντικός παράγοντας για σύγκριση με τους παραδοσιακούς αλγόριθμους της Συνδυαστικής Βελτιστοποίησης. Ένα άλλο πρόβλημα που πρέπει να αντιμετωπιστεί είναι η ανάπτυξη συγκεκριμένων αλγορίθμων και στρατηγικών εκπαίδευσης για τη βελτίωση της γενίκευσης, δηλαδή εκπαίδευση σε μικρότερες περιπτώσεις προβλημάτων και γενίκευση σε μεγαλύτερες κλίμακες. Στην ίδια γραμμή έρευνας, θα πρέπει να διερευνηθεί η γενίκευση σε άλλες περιπτώσεις προβλημάτων με διαφορετικές κατανομές. Η εργασία για την επινόηση πιο γενικών αλγορίθμων που μπορούν να δουλέψουν με διάφορες κατηγορίες συνδυαστικών προβλημάτων και με διατυπώσεις συγκεκριμένων προβλημάτων είναι επίσης μια πολλά υποσχόμενη ερευνητική κατεύθυνση. Τέλος, οι τρέχουσες προσεγγίσεις χρησιμοποιούν συχνά τις βασικές παραλλαγές των αλγορίθμων Ενισχυμένης Μάθησης, επομένως η χρήση των πιο σύγχρονων προσεγγίσεων από το πεδίο της CO θα μπορούσε να αποδειχθεί ευεργετική στο μέλλον λόγω της αυξημένης απόδοσης και σταθερότητας του δείγματος, καθώς και της ενσωμάτωσης αποτελεσματικότερων τεχνικών εκμάθησης των ζητούμενων αναπαραστάσεων.

Επιπλέον, η διερεύνηση για τη βελτίωση γενικών ευρετικών που μπορούν να εφαρμοστούν σε μεγάλο αριθμό συνδυαστικών προβλημάτων είναι μια άλλη ενδιαφέρουσα ιδέα για περαιτέρω ανάπτυξη. Ένα μειονέκτημα της μεθόδου Policy Gradient που περιγράψαμε στα προηγούμενα κεφάλαια, είναι ο μεγάλος αριθμός δειγμάτων που απαιτούνται για την κατάρτιση μιας καλής πολιτικής. Ως μελλοντική κατεύθυνση, προτείνεται η εξερεύνηση μεθόδων που μπορούν να είναι πιο αποτελεσματικές ως δείγμα και να μάθουν καλές πολιτικές που απαιτούν λιγότερο χρόνο εκπαίδευσης.

Στα πλαίσια μελλοντικής έρευνας θα μπορούσε να διερευνηθεί η έννοια και η δομή του Meta-Reinforcement Learning. Στις σύγχρονες μέρες της Βαθιάς Μάθησης, οι Wang et al., (2016) και Duan et al., (2017) πρότειναν ταυτόχρονα την πολύ παρόμοια ιδέα του Meta-RL (ονομάζεται RL^2 στη δεύτερη ερευνητική εργασία). Ένα μοντέλο meta-RL εκπαιδεύεται σε μια κατανομή MDP και κατά τη δοκιμή, είναι σε θέση να μάθει να επιλύει γρήγορα μια νέα εργασία. Ο στόχος του meta-RL είναι φιλόδοξος, κάνοντας ένα βήμα παραπέρα προς τους γενικούς αλγόριθμους. Το Meta-RL είναι μετα-μάθηση σε εργασίες Ενισχυμένης Μάθησης. Αφού εκπαιδευτεί σε μια κατανομή εργασιών, ο agent είναι σε θέση να λύσει μια νέα εργασία αναπτύσσοντας έναν νέο αλγόριθμο RL εκμεταλλευόμενος τη δυναμική της εσωτερικής δραστηριότητας του. Για να ανακεφαλαιώσουμε, ένα καλό μοντέλο μετα-μάθησης καλείται να γενικευτεί σε νέες εργασίες ή νέα περιβάλλοντα που δεν έχουν συναντηθεί ποτέ κατά τη διάρκεια της εκπαίδευσης. Η διαδικασία προσαρμογής, ουσιαστικά αποτελεί μια μίνι συνεδρία μάθησης και πραγματοποιείται κατά τη δοκιμή με περιορισμένη έκθεση στις νέες διαμορφώσεις. Ακόμη και χωρίς ρητή μικρο-ρύθμιση, το μοντέλο μετα-μάθησης προσαρμόζει αυτόνομα εσωτερικές κρυφές καταστάσεις για μάθηση.

Επιπρόσθετα, αξίζει να διερευνηθεί και η βιβλιοθήκη OR-Gym (Hubbs C. et al., 2020). Η OR-Gym είναι μια βιβλιοθήκη ανοιχτού κώδικα για την ανάπτυξη αλγορίθμων Ενισχυμένης Μάθησης ώστε να αντιμετωπίσουν τα προβλήματα της επιχειρησιακής έρευνας. Η βιβλιοθήκη αυτή, αναπτύσσει περιβάλλοντα που βασίζονται σε πρωτότυπα μοντέλα στη βιβλιογραφία και εφαρμόζουν διάφορα μοντέλα βελτιστοποίησης και ευρετικές συναρτήσεις, προκειμένου να συγκριθούν τα αποτελέσματα RL. Αναπλαισιώνοντας, μια σειρά από κλασικά προβλήματα βελτιστοποίησης ως εργασίες RL, επιδιώκουν να παρέχουν ένα νέο εργαλείο για την κοινότητα της επιχειρησιακής έρευνας, ενώ παράλληλα ανοίγουν το δρόμο για την κοινότητα του RL σε πολλά από τα προβλήματα και τις προκλήσεις στον τομέα OR.

Καταλήγοντας, αξίζει να αναφερθεί και το ερευνητικό πλαίσιο Dopamine (Castro PS et al., 2018). Η έρευνα για τη βαθιά Ενισχυμένη Μάθηση (deep RL) έχει αυξηθεί σημαντικά τα τελευταία χρόνια. Υπάρχουν πλέον διάφορες προσφορές λογισμικού που παρέχουν σταθερές, ολοκληρωμένες υλοποιήσεις προς συγκριτική αξιολόγηση. Ταυτόχρονα, η πρόσφατη έρευνα σε μεθόδους DRL έχει διαφοροποιηθεί ως προς τους

πρωταρχικούς στόχους της. Το Dopamine ένα νέο ερευνητικό πλαίσιο για βαθιά RL που στοχεύει να υποστηρίξει ένας μέρος της ποικιλομορφίας και της διαφορετικότητας που αναφέρεται παραπάνω. Πιο συγκεκριμένα, είναι ένα ερευνητικό πλαίσιο είναι ανοιχτού κώδικα, βασίζεται στη βιβλιοθήκη TensorFlow και παρέχει συμπαγείς και αξιόπιστες υλοποιήσεις ορισμένων agents που υπόκεινται σε μεθόδους Deep Reinforcement Learning τελευταίας τεχνολογίας.

Τέλος, οι ειδικοί πιστεύουν ότι η βαθιά Ενισχυμένη Μάθηση βρίσκεται στην αιχμή της αυτή τη στιγμή και επιτέλους έφτασε στο σημείο ώστε να μπορεί να παρέχει αξιόπιστα αποτελέσματα σε πραγματικές εφαρμογές. Πιστεύουν επίσης, ότι η εδραίωση και η περαιτέρω ανάπτυξή της θα έχει μεγάλο αντίκτυπο στην πρόοδο της τεχνητής νοημοσύνης και μπορεί τελικά οι ερευνητές να πλησιάσουν ακόμα περισσότερο την Γενική Τεχνητή Νοημοσύνη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Christos H Papadimitriou, 'The euclidean travelling salesman problem is np-complete', *Theoretical Computer Science*, 4(3), 237–244, (1977).
2. Larson, R. C., & Odoni, A. R. (1981). *Urban operations research* (No. Monograph).
3. Wang, H., Ma, C., & Zhou, L. (2009, December). A brief review of machine learning and its application. In *2009 international conference on information engineering and computer science* (pp. 1-4). IEEE.
4. Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 105400.
5. Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. (2016). Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*.
6. Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. *arXiv preprint arXiv:1704.01665*.
7. Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, *290*(2), 405-421.
8. Ma, Q., Ge, S., He, D., Thaker, D., & Drori, I. (2019). Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. *arXiv preprint arXiv:1911.04936*.
9. Deudon, M., Cournut, P., Lacoste, A., Adulyasak, Y., & Rousseau, L. M. (2018, June). Learning heuristics for the tsp by policy gradient. In *International conference on the integration of constraint programming, artificial intelligence, and operations research* (pp. 170-181). Springer, Cham.
10. da Costa, P. R. D. O., Rhuggenaath, J., Zhang, Y., & Akcay, A. (2020). Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning. *arXiv preprint arXiv:2004.01608*.
11. La Maire, B. F., & Mladenov, V. M. (2012, September). Comparison of neural networks for solving the travelling salesman problem. In *11th Symposium on Neural Network Applications in Electrical Engineering* (pp. 21-24). IEEE.

12. M. Junger, S. Thienel, and G. Reinelt, “Provably good solutions for the traveling salesman problem,” *Mathematical Methods of Operations Research*, vol. 40, pp. 183–217, 1994.
13. N. K. Bose and P. Liang, *Neural Network Fundamentals with Graphs, Algorithms, and applications*, ser. McGraw-Hill Electrical and Computer Engineering Series. McGraw-Hill, Inc., 1996.
14. G.Laporte,“The traveling salesman problem: An overview of exact and approximate algorithms,” *European Journal of Operational Research*, vol. 59, pp. 2331–247, 1992.
15. I. K. Altinel, N. Aras, and B. J. Oommen, “Fast, effcient and accurate solutions to the hamiltonian path problem using neural approaches,” *Computers & Operations Research*, vol. 27, pp. 461–494, 2000.
16. J. J. Hopfield and D. W. Tank, ““neural” computation of decisions in optimization problems,” *Biological Cybernetics*, vol. 52, pp. 141–152, 1985.
17. F. Sarwar and A. A. Bhatti, “Critical analysis of hopfield’s neural network model for tsp and its comparison with heuristic algorithm for shortest path computation,” in *Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST)*, 2012.
18. C. Lau and B. Widrow, Eds., *Special Issue on Neural Networks*, ser. Proceedings IEEE, vol. 78, September/October 1990.
19. R. Woodburn and A. Murray, *Neural Network Analysis, Architecture and Applications*, 1997, ch. Pulse-Stream Techniques and Circuits for implementing Neural Netorks.
20. F.-L. Luo and R.Unbehauen, *Applied Neural Network for Signal Processing*. Cambridge University Press, 1998.
21. D. Zhang, *Parallel VLSI Neural System Design*. Verlag, 1999.
22. Christofides, N. (1976). The vehicle routing problem. *Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle*, *10*(V1), 55-70.
23. David Applegate, Robert Bixby, Vasek Chvatal, and William Cook. Implementing the dantzig-fulkerson-johnson algorithm for large traveling salesman problems. *Mathematical programming*, 2003.
24. David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook. Concorde tsp solver, 2006. URL www.math.uwaterloo.ca/tsp/concorde.

25. David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook. *The traveling salesman problem: a computational study*. Princeton university press, 2011.
26. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
27. Nicos Christofides. Worst-case analysis of a new heuristic for the Travelling Salesman Problem. In *Report 388*. Graduate School of Industrial Administration, CMU, 1976.
28. Fred Glover and Manuel Laguna. *Tabu Search*. Springer, 2013.
29. Google Or-tools, google optimization tools, 2016. URL https://developers.google.com/optimization/routing.
30. Keld Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman. *European Journal of Operational Research*, 126:106–130, 2000.
31. Keld Helsgaun. LK-H, 2012. URL http://akira.ruc.dk/˜keld/research/LKH/.
32. S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.
33. Christos H. Papadimitriou. The Euclidean Travelling Salesman Problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.
34. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57-81.
35. Vesselinova, N., Steinert, R., Perez-Ramirez, D. F., & Boman, M. (2020). Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access*, *8*, 120388-120416.
36. C. Han, S. Tang, M. Ding, Solving combinatorial problems with machine learning methods, in: Nonlinear Combinatorial Optimization, Springer, 2019, pp. 207–229.
37. Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., ... & Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.
38. Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., & Abbeel, P. (2016). RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.
39. Hubbs, C. D., Perez, H. D., Sarwar, O., Sahinidis, N. V., Grossmann, I. E., & Wassick, J. M. (2020). OR-Gym: A Reinforcement Learning Library for Operations Research Problems. *arXiv preprint arXiv:2008.06319*.

40. Castro, P. S., Moitra, S., Gelada, C., Kumar, S., & Bellemare, M. G. (2018). Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*.