

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΧΕΔΙΟ ΠΑΡΟΥΣΙΑΣΗΣ
ΑΤΟΜΙΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Μάιος 2023

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΑΛΓΟΡΙΘΜΩΝ
ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΤΗΝ ΤΟΠΟΘΕΤΗΣΗ
ΔΕΔΟΜΕΝΩΝ ΣΕ ΕΤΕΡΟΓΕΝΗ ΣΥΣΤΗΜΑΤΑ ΜΝΗΜΗΣ**

Χαράλαμπος Λαμπρή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη και αξιολόγηση αλγορίθμων μηχανικής μάθησης για την τοποθέτηση
δεδομένων σε ετερογενή συστήματα μνήμης**

Χαράλαμπος Λαμπρή

Επιβλέπων Καθηγητής
Χάρης Βώλος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων
απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου
Κύπρου

Μάιος 2023

Ευχαριστίες

Θα ήθελα να εκφράσω την βαθιά μου ευγνωμοσύνη προς τον επιβλέποντα Καθηγητή μου, Δρ. Χάρη Βώλο, για τη συνεχή υποστήριξη, την καθοδήγηση και τις πολύτιμες συμβουλές του καθ' όλη τη διάρκεια της πραγματοποίησης αυτής της διπλωματικής εργασίας.

Θα ήθελα επίσης να ευχαριστήσω τους συμφοιτητές μου για τις χρήσιμες συζητήσεις, τις πολύτιμες παρατηρήσεις και τη γενική βοήθεια καθ' όλη τη διάρκεια της ερευνητικής μου εργασίας.

Τέλος, είμαι ευγνώμον προς την οικογένειά μου και τους φίλους μου για την αγάπη, την κατανόηση και την υποστήριξη τους, ιδιαίτερα στις πιο δύσκολες στιγμές. Χωρίς την υποστήριξη και την ενθάρρυνσή τους, αυτό το εγχείρημα δεν θα ήταν εφικτό.

Περίληψη

Η ευρεία ενσωμάτωση των ετερογενών τεχνολογιών μνήμης στα σύγχρονα υπολογιστικά συστήματα είναι ευρέως αναγνωρισμένη. Αυτή η ενοποίηση εξυπηρετεί τη συνεχώς κλιμακούμενη απαίτηση για μεγαλύτερα μεγέθη κύριας μνήμης που τίθενται από τις αναδυόμενες εφαρμογές. Αυτές οι νέες τεχνολογίες παρουσιάζουν σημαντικές διαφορές ως προς το μέγεθος, την καθυστέρηση πρόσβασης και το εύρος ζώνης. Η ετερογένεια αυτών των συστημάτων, σε συνδυασμό με τα μοναδικά χαρακτηριστικά των νέων εφαρμογών, καθιστούν τις τρέχουσες πρακτικές διαχείρισης μνήμης ανεπαρκείς.

Σε αυτή την ερευνητική διπλωματική εργασία, σχεδιάσαμε και αξιολογήσαμε έναν έξυπνο δρομολογητή που τοποθετεί δεδομένα εφαρμογής σε επίπεδο σελίδας σε διάφορα στοιχεία μνήμης του συστήματος χρησιμοποιώντας τεχνητά νευρωνικά δίκτυα. Ο προτεινόμενος δρομολογητής αξιοποιεί τα επαναλαμβανόμενα νευρωνικά δίκτυα σε συνδυασμό με τις υπάρχουσες μεθόδους τοποθέτησης δεδομένων που βασίζονται σε ιστορικά δεδομένα. Ο δρομολογητής μας χρησιμοποιεί έξυπνα τη μηχανική μάθηση για ένα υποσύνολο σελίδων που, όταν μετακινηθούν την κατάλληλη στιγμή στην κατάλληλη μονάδα μνήμης, επιτυγχάνουν σημαντική αύξηση της απόδοσης εφαρμογής.

Αναπτύξαμε επίσης μοντέλα ARIMA για τον ίδιο στόχο με τα RNN που αναφέραμε στην πιο πάνω παράγραφο. Κατόπιν αυτού, πραγματοποιήθηκε μια συγκριτική ανάλυση των μοντέλων ARIMA και RNN, παρέχοντας περαιτέρω πληροφορίες για τις αντίστοιχες ικανότητες και επιδόσεις τους.

Τέλος βαθμολογήσαμε την αποδοτικότητα των προαναφερθέντων μοντέλων που χρησιμοποιήθηκαν από τον δρομολογητή μας. Αυτό διεκπεραιώθηκε μέσω μιας προγραμματιστικής προσημείωσης υβριδικού συστήματος μνήμης που αποτελούταν από δύο μονάδες μνήμης την DRAM και την NVM. Κατά συνέπια δείξαμε ότι είναι εφικτή η καλύτερη απόδοση μιας εφαρμογής ακόμη και όταν διαχειριζόμαστε μερικές σελίδες από το συνολικό ίχνος μιας εφαρμογής με έξυπνους τρόπους.

Λέξεις κλειδιά - Ετερογενή Συστήματα Μνήμης, Μηχανική Μάθηση, Νευρωνικά Δίκτυα, RNN, LSTM, DRAM, NVM, Επαναλαμβανόμενα Νευρωνικά Δίκτυα, Δρομολογητής Σελίδων, Στατιστικά Μοντέλα, ARIMA, AR, I, MA.

Περιεχόμενα

Κεφάλαιο 1 - Εισαγωγή	1
1.1 Δήλωση Γενικού Προβλήματος	1
1.2 Αντικείμενο Μελέτης	2
1.3 Κίνητρο και Συνεισφορά	4
1.4 Μεθοδολογία	6
Κεφάλαιο 2 - Υπόβαθρο Μηχανικής Μάθησης	7
2.1 Ορισμός	7
2.2 Είδη Μηχανικής Μάθησης	7
2.3 Τεχνητά Νευρωνικά Δίκτυα	9
2.4 Επαναλαμβανόμενα Νευρωνικά Δίκτυα	11
2.5 LSTMs	13
2.6 Συναρτήσεις Ενεργοποίησης	14
2.7 Συνάρτηση Κόστους	15
2.8 Τεχνικές Βελτιστοποίησης και Αλγόριθμοι	16
Κεφάλαιο 3 - Δρομολογητής Σελίδων	19
3.1 Εισαγωγή	19
3.2 Επισκόπηση Λειτουργίας Δρομολογητή Συστήματος Υβριδικής Μνήμης	20
3.2.1 Ενέργεια πρώτη – Επιλογέας Σελίδας	20
3.2.2 Ενέργεια δεύτερη – Έξυπνος τρόπος διαχείρισης τοποθέτησης σελίδων	22
3.2.3 Ενέργεια τρίτη – Εκχώρηση σελίδων στις μονάδες μνήμης	23
3.3 Χρήση πολιτικής αδειάσματος DRAM από τον δρομολογητή	23
Κεφάλαιο 4 - Έξυπνη Διαχείριση τοποθέτησης σελίδων με RNN	24
4.1 Εισαγωγή	24
4.2 Προσέγγιση με Επαναλαμβανόμενα Νευρωνικά Δίκτυα	24
4.3 Είσοδος στα Επαναλαμβανόμενα Νευρωνικά Δίκτυα	25
4.4 Πρώτη Προσέγγιση – Πρόβλεψη σελίδας που θα αντικατασταθεί από DRAM	26
4.5 Δεύτερη Προσέγγιση – Πρόβλεψη πότε θα αντικατασταθεί η κάθε σελίδα	27
4.6 Τεχνική Υλοποίηση	28
Κεφάλαιο 5 - Έξυπνη Διαχείριση τοποθέτησης σελίδων με ARIMA(p,d,q)	30
5.1 Εισαγωγή	30
5.2 Υπόβαθρο χρονολογικών σειρών	31
5.2.1 Ορισμός Χρονολογικής σειράς	31
5.2.2 Στασιμότητα	32
5.3 Υπόβαθρο μοντέλων ARIMA(p,d,q)	33

5.3.1 AR(p) - Αυτοπαλινδρομικό Μοντέλο	33
5.3.2 I(d) – Όρος Διαφοροποίησης	34
5.3.3 MA(q) - Μοντέλο κινούμενου μέσου όρου πεπερασμένης τάξης q	34
5.3.4 Επιλογή παραμέτρων p, d, q	34
5.4 Προσέγγιση – Πρόβλεψη πότε θα αντικατασταθεί η κάθε σελίδα	35
5.5 Τεχνική Υλοποίηση.....	36
Κεφάλαιο 6 - Αξιολόγηση Αποτελεσμάτων	38
6.1 Εισαγωγή.....	38
6.2 Ίχνη εφαρμογών σαν Δεδομένα Εισόδου.....	39
6.3 Προσομοιωτής Συστήματος Υβριδικής Μνήμης	39
6.4 Ακρίβεια πρόβλεψης μοντέλων RNNs	40
6.5 Ακρίβεια πρόβλεψης μοντέλων ARIMA's	41
6.6 Απόδοση Εφαρμογών με χρήση RNNs.....	42
6.7 Απόδοση Εφαρμογών με χρήση ARIMA	44
Κεφάλαιο 7 - Συμπεράσματα.....	46
7.1 Συμπεράσματα έρευνας.....	46
7.2 Προτάσεις για μελλοντική έρευνα	47
Βιβλιογραφία	48
Παράρτημα Α.....	A-2
Παράρτημα Β.....	B-2

Κεφάλαιο 1

Εισαγωγή

1.1 Δήλωση Γενικού Προβλήματος	1
1.2 Αντικείμενο μελέτης	2
1.3 Κίνητρο και Συνεισφορά	4
1.4 Μεθοδολογία	6

1.1 Δήλωση Γενικού Προβλήματος

Είναι πλέον προφανές ότι η μηχανική μάθηση έχει γίνει αναπόσπαστο κομμάτι διαφόρων τομέων, διευκολύνοντας μας στην επίλυση πολύπλοκων προβλημάτων. Η χρήση μηχανικής μάθησης επέτρεψε την ανάπτυξη καινοτόμων λύσεων σε ένα ευρύ φάσμα εφαρμογών. Αυτό είχε ως αποτέλεσμα την τεράστια ανάπτυξη και πρόοδο στον τομέα της τεχνητής νοημοσύνης.

Οι πρόσφατες εξελίξεις στην αρχιτεκτονική των υπολογιστών, όπως η χρήση Field-Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPU) και Application-Specific Integrated Circuits (ASIC) έχουν ανοίξει νέες δυνατότητες για τη βελτίωση της απόδοσης των αλγορίθμων μηχανικής εκμάθησης [11]. Αυτοί οι εξειδικευμένοι επιταχυντές υλικού μπορούν να επιταχύνουν σημαντικά τους υπολογισμούς που απαιτούνται για την εκπαίδευση και την εξαγωγή συμπερασμάτων, καθιστώντας δυνατή την επεξεργασία τεράστιων ποσοτήτων δεδομένων σε πραγματικό χρόνο.

Παρά την ταχεία πρόοδο στην τεχνητή νοημοσύνη, υπήρξε μια αισθητή επιβράδυνση στην πρόοδο της αρχιτεκτονικής των υπολογιστών. Το χάσμα μεταξύ της κεντρικής μονάδας επεξεργασίας (Central processing unit, CPU) και των μονάδων μνήμης συνεχίζει να αυξάνεται και τα μοτίβα βελτίωσης της απόδοσης που προβλέπονται από το νόμο του Moore φαίνεται να αγγίζουν τα όριά τους.

Για την αντιμετώπιση αυτού του ζητήματος, είναι απαραίτητο να υιοθετηθεί μια πιο αλληλένδετη προσέγγιση όπου η τεχνητή νοημοσύνη και ο σχεδιασμός της αρχιτεκτονικής υπολογιστών μπορούν να αλληλοσυμπληρώνονται. Αυτή η προσέγγιση περιλαμβάνει τη αξιοποίηση των δυνατοτήτων και των δύο τομέων για να ξεπεραστούν οι περιορισμοί και οι προκλήσεις που αντιμετωπίζει ο καθένας τους. Για παράδειγμα, η τεχνητή νοημοσύνη μπορεί να βοηθήσει στη βελτιστοποίηση του σχεδιασμού της αρχιτεκτονικής υπολογιστών, εντοπίζοντας τα σημεία συμφόρησης στην απόδοση και προτείνοντας τρόπους βελτίωσης της αποδοτικότητας του συστήματος. Ομοίως, η αρχιτεκτονική των υπολογιστών μπορεί να βοηθήσει στην ανάπτυξη πιο αποδοτικών και αποτελεσματικών αλγορίθμων τεχνητής νοημοσύνης παρέχοντας εξειδικευμένο υλικό, βελτιστοποιημένο για συγκεκριμένες εργασίες.

Συμπερασματικά, η αυξανόμενη τάση εισαγωγής λύσεων μηχανικής μάθησης σε διάφορους τομείς έχει διευκολύνει την τεράστια ανάπτυξη στον τομέα της τεχνητής νοημοσύνης. Ωστόσο, οι επίμονες προκλήσεις που αντιμετωπίζει ο σχεδιασμός της αρχιτεκτονικής υπολογιστών πρέπει να αντιμετωπιστούν για να συμβαδίσουν με τις αυξανόμενες απαιτήσεις των αλγορίθμων μηχανικής μάθησης. Η αλληλένδετη χρήση της τεχνητής νοημοσύνης και του σχεδιασμού της αρχιτεκτονικής υπολογιστών μπορεί να προσφέρει καινοτόμες λύσεις για να ξεπεραστούν αυτές οι προκλήσεις και να διευκολυνθεί η περαιτέρω ανάπτυξη και πρόοδος στον τομέα.

1.2 Αντικείμενο Μελέτης

Η παρούσα διπλωματική εργασία επικεντρώνεται γύρω από τη χρήση μεθόδων μηχανικής μάθησης και μαθηματικών μοντέλων με στόχο την βελτίωση των τεχνικών δρομολογητών σελίδων όπου τοποθετούν δεδομένα σε ιεραρχία μνήμης που περιλαμβάνει ετερογενείς τεχνολογίες μνήμης και αποθήκευσης. Νέες τεχνολογίες όπως η μη-πτητική μνήμη (Non-volatile memory, NVM) και η απομακρυσμένη μνήμη (Remote memory) εισάγουν νέα επίπεδα στην παραδοσιακή ιεραρχία μνήμης. Σε σύγκριση με την παραδοσιακή τεχνολογία μνήμης DRAM, τα νέα αυτά επίπεδα στην ιεραρχία μνήμης προσφέρουν επιλογές χαμηλότερου κόστους, μεγαλύτερης χωρητικότητας, αλλά χαμηλότερης επίδοσης [1]. Επομένως τα νέα επίπεδα μνήμης δίνουν νέες δυνατότητες για την βελτιστοποίηση των εφαρμογών στα σύγχρονα υπολογιστικά συστήματα. Αυξάνοντας όμως τις επιλογές τοποθέτησης, η βελτιστοποίηση για το ποια δεδομένα πρέπει να τοποθετηθούν σε κάθε επίπεδο μνήμης γίνεται πιο πολύπλοκη.

Το κυρίως αντικείμενο που ασχολείται η συγκεκριμένη διπλωματική εργασία εστιάζει στην διερεύνηση καταλληλότητας και αποτελεσματικότητας μεθόδων στηριζόμενων σε μηχανική μάθηση και μαθηματική μοντελοποίηση για πρόβλεψη χρονολογικών σειρών (time series) του ενδιαφέροντος δρομολογητή σελίδων κάτω από ένα συγκεκριμένο σενάριο. Το σενάριο είναι όταν έχουμε ένα σύγχρονο σύστημα υπολογιστών που χρησιμοποιεί ένα υβριδικό σύστημα μνήμης που αποτελείται από την Δυναμική μνήμη τυχαίας προσπέλασης (Dynamic Random Access Memory, DRAM) και την Μόνιμη Μνήμη (Persistent Memory, PM) που χρησιμοποιείται ως αύξηση της χωρητικότητάς της κύριας μνήμης.

Είναι κοινώς αποδεκτό ότι τα σύγχρονα συστήματα υπολογιστών σχεδιάζονται χρησιμοποιώντας ετερογενή στοιχεία μνήμης για να επεκτείνουν τη χωρητικότητα της κύριας μνήμης ή για σκοπούς προσωρινής αποθήκευσης. Αυτά τα ετερογενή συστήματα μνήμης έχουν συμβιβασμούς που πρέπει να αντιμετωπιστούν, όπως η αυξημένη καθυστέρηση και το μειωμένο εύρος ζώνης σε στοιχεία μνήμης που βρίσκονται πιο μακριά από την υπολογιστική μονάδα.

Ένας δρομολογητής σελίδων (Page Scheduler) είναι το επίπεδο διαχείρισης μνήμης των συστημάτων λειτουργίας και του χρόνου εκτέλεσης που είναι υπεύθυνα για τη μετεγκατάσταση σελίδων σε ετερογενή στοιχεία μνήμης. Ο στόχος ενός αποτελεσματικού δρομολογητή σελίδων είναι να διασφαλίσει ότι οι σελίδες με συχνή πρόσβαση είναι άμεσα διαθέσιμες στις μονάδες μνήμης υψηλής απόδοσης (DRAM), ενώ οι λιγότερο σημαντικές σελίδες αποθηκεύονται σε μια πιο αργή μονάδα μνήμης (Non-volatile memory, NVM).

Είναι σημαντικό να σημειωθεί ότι τα υβριδικά συστήματα μνήμης γίνονται όλο και πιο δημοφιλή στα σύγχρονα συστήματα υπολογιστών λόγω της ικανότητάς τους να προσφέρουν υψηλή χωρητικότητα, χαμηλή καθυστέρηση και χαμηλή κατανάλωση ενέργειας. Η PM είναι ένας σχετικά νέος τύπος μνήμης που κερδίζει έδαφος στον κλάδο καθώς παρέχει μη πτητική μνήμη με δυνατότητα διευθυνσιοδότησης δυφιοσυλλαβής (byte) και έχει παρόμοιους χρόνους πρόσβασης με τη DRAM. Η PM χρησιμοποιείται συνήθως ως επέκταση της κύριας μνήμης σε υβριδικά συστήματα μνήμης [1], επιτρέποντας στα συστήματα να αυξάνουν τη χωρητικότητα της μνήμης τους χωρίς την ανάγκη δαπανηρής αναβάθμισης DRAM. Ωστόσο, καθώς η PM είναι πιο αργή από την DRAM, απαιτείται ένας αποτελεσματικός δρομολογητής σελίδων για τη διαχείριση της μετεγκατάστασης δεδομένων μεταξύ των δύο τύπων μνήμης.

Τα νευρωνικά δίκτυα έχουν αποδειχθεί αποτελεσματικά στην επίλυση πολύπλοκων προβλημάτων βελτιστοποίησης και, ως εκ τούτου, μπορούν να αξιοποιηθούν για να σχεδιαστεί ένας αποτελεσματικός δρομολογητής σελίδων για υβριδικά συστήματα μνήμης. Λόγω του ότι η εκπαίδευση νευρωνικών δικτύων συνήθως είναι χρονοβόρα είναι επιτακτικό να διερευνήσουμε αν υπάρχουν άλλα μαθηματικά μοντέλα και αλγόριθμοι που μπορούν να προσφέρουν επίλυση πολύπλοκων προβλημάτων βελτιστοποίησης σε ικανοποιητικό χρόνο.

1.3 Κίνητρο και Συνεισφορά

Πολλοί ερευνητές έχουν προσπαθήσει να αντιμετωπίσουν το δύσκολο πρόβλημα της οργάνωσης δεδομένων σε επίπεδο σελίδων εφαρμογών και της κατανομής τους στα κατάλληλα στοιχεία μνήμης σε ένα σύστημα. Αυτή η εργασία είναι περίπλοκη επειδή απαιτεί την εξέταση της περίπλοκης αλληλεπίδρασης μεταξύ των μοτίβων πρόσβασης των σελίδων της εφαρμογής. Οι ερευνητές έχουν προτείνει λύσεις [2,3,15,19] που μπορούν να ενσωματωθούν σε διάφορα επίπεδα του συστήματος, συμπεριλαμβανομένου υλικού, μεταγλωττιστών, λειτουργικών συστημάτων και περιβαλλόντων χρόνου εκτέλεσης. Παρά τις διαφορές μεταξύ αυτών των προσεγγίσεων, μοιράζονται μια κοινή εξάρτηση από ιστορικά δεδομένα σχετικά με τις προσβάσεις σε σελίδες εφαρμογών. Οι τρέχουσες λύσεις δυναμικής διαχείρισης σελίδων τελευταίας τεχνολογίας για ετερογενή συστήματα μνήμης χρησιμοποιούν άμεση παρατηρούμενη συμπεριφορά για να καθορίσουν την καλύτερη μελλοντική τοποθέτηση σελίδας. Ωστόσο, αυτή η στρατηγική είναι ανεπαρκής για την αποτύπωση των περίπλοκων προτύπων πρόσβασης των σύγχρονων εφαρμογών. Αργότερα σε αυτή τη διπλωματική, θα δείξουμε γιατί αυτή η απλοϊκή πολιτική (πολιτική ιστορικού) απέχει πολύ από τη βέλτιστη [2,3] (πολιτική oracle).

Η παρούσα διπλωματική εργασία στοχεύει στη διερεύνηση και την κατασκευή ενός δρομολογητή σελίδων σε ένα ειδικά σχεδιασμένο υβριδικό σύστημα μνήμης [3] που χρησιμοποιεί μηχανική μάθηση. Ο πρωταρχικός στόχος είναι να γεφυρωθεί το χάσμα απόδοσης μεταξύ της τρέχουσας προσέγγισης που βασίζεται στην πολιτική ιστορικού και του ιδανικού αλλά ανέφικτου δρομολογητή σελίδων Oracle. Η εστίαση είναι στην παροχή μιας σχεδόν βέλτιστης τοποθέτησης δεδομένων σε ετερογενή στοιχεία μνήμης σε ευκρίνεια σελίδας.

Η διπλωματική στοχεύει στην πραγματοποίηση πολλών συγκεκριμένων θεμάτων. Πρώτον, θα διερευνήσει το κενό απόδοσης στις τρέχουσες λύσεις, καθώς οι πρόσφατες τεχνολογίες όπως η Persistent Memory δεν έχουν δώσει ακόμη αρκετό χρόνο για την ανάπτυξη πολλών εφαρμόσιμων αλγορίθμων. Η προσέγγιση που βασίζεται στην ιστορία είναι σχετικά αφελής αλγοριθμικά εφόσον το μόνο που κάνει είναι να θεωρεί ότι ο αριθμός προσβάσεων της σελίδας x την χρονική στιγμή t θα είναι ο ίδιος και την χρονική στιγμή $t+1$. Αυτό συνεπάγεται με το ότι δεν είναι ικανή να καταγράψει πολύπλοκα μοτίβα πρόσβασης.

Δεύτερον, η διατριβή θα διερευνήσει τη χρήση της τεχνητής νοημοσύνης που βασίζεται στη μηχανική μάθηση, και συγκεκριμένα των επαναλαμβανόμενων νευρωνικών δικτύων (RNNs). Η διατριβή θα προσδιορίσει τα RNNs ως μια αποτελεσματική και πρακτική τεχνική για το πρόβλημα του προγραμματισμού σελίδων (δρομολόγηση σελίδων), όπως τεκμηριώνεται στα άρθρα [2,3]. Θα γίνουν οι απαραίτητες προσαρμογές και προσπάθειες για την επίτευξη υψηλής ακρίβειας μοντέλων RNN. Η διατριβή θα διαπιστώσει ότι η εκπαίδευση ενός RNN ανά σελίδα μπορεί να οδηγήσει σε υψηλή ακρίβεια και σημαντικές βελτιώσεις απόδοσης, ακόμη και όταν εφαρμόζεται σε ένα σχετικά μικρό υποσύνολο σελίδων εφαρμογής. Με το κύριο αρνητικό να είναι ότι ο χρόνος εκπαίδευσης των RNN είναι αρκετά υψηλός και μη πρακτικός. Έτσι θα προσπαθήσουμε να τον αντιμετωπίσουμε με την εισαγωγή συγκεκριμένων μεταβλητών-αριθμών που θα ορίζουν την εποχική δρομολόγηση σελίδων σύμφωνα με το άρθρο [4], περισσότερη επεξήγηση θα ακολουθήσει στα επόμενα κεφάλαια.

Τρίτον, η διατριβή θα προσπαθήσει να προτείνει ένα πιο γρήγορο και αποτελεσματικό τρόπο εκπαίδευσης (στατιστικά μοντέλα ARIMA) σε σύγκριση με τα μοντέλα RNNs ανά σελίδα που θα οδηγεί και αυτός με την σειρά του σε ένα αποδεκτό ύψος ακρίβειας απόδοσης με κύριο στόχο να είναι όσο το δυνατό ταχύτερος.

Τέλος, η διατριβή θα σχεδιάσει έναν δρομολογητή σελίδων που ακολουθεί παρόμοια υλοποίηση με τον δρομολογητή ονόματος 'KLEIO' που αναπτύχθηκε στο άρθρο [3] αφού ληφθούν υπόψη διάφορες μετρήσεις απόδοσης.

Πιο συνοπτικά η προσέγγιση του δρομολογητή σελίδων θα συνδυάζει τόσο την πολιτική που βασίζεται στην ιστορία όσο και τη μηχανική μάθηση αλλά και μαθηματικών μοντέλων (ARIMA). Η υλοποίηση γίνεται με τη χρήση RNN, και συγκεκριμένα τη Μακροπρόθεσμη Βραχυπρόθεσμη Μνήμη (LSTM). Οι βελτιώσεις απόδοσης που επιτυγχάνονται θα μετρηθούν χρησιμοποιώντας μια σειρά φόρτων εργασίας που χρησιμοποιήθηκαν στα άρθρα [2,3]. Η αξιολόγηση του δρομολογητή σελίδων θα περιστρέφεται τόσο γύρω από την

πραγματική απόδοση (DRAM hit rate) καθώς και τα ποσοστά ακριβών προβλέψεων από έξυπνους τρόπους διαχείρισης τοποθέτησης σελίδων σε ένα υβριδικό σύστημα που θα αποτελείται από την DRAM και την NVM.

Συνολικά, αυτή η διπλωματική στοχεύει να συνεισφέρει στον τομέα του σχεδιασμού του Υβριδικού Συστήματος Μνήμης διερευνώντας και κατασκευάζοντας έναν δρομολογητή σελίδων που χρησιμοποιεί τη μηχανική μάθηση και μαθηματικά στατιστικά μοντέλα για να παρέχει όσο το δυνατό καλύτερη τοποθέτηση δεδομένων σε ετερογενή στοιχεία μνήμης.

1.4 Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε για την εκπόνηση αυτής της διπλωματικής εργασίας ήταν πρώτιστος η εύρεσης πηγών (βιβλιογραφία) που θα αποτελούσε την βάση για να στηριχθεί η συγκεκριμένη έρευνα. Αυτό επιτεύχθηκε με την αναζήτηση λέξεων κλειδιών που αναφέραμε στο κομμάτι της περίληψης στην σελίδα 5 αυτής της διπλωματικής με κύριο στόχο την εύρεση επιστημονικών άρθρων, περιοδικών, βιβλίων και ιστοσελίδων. Όταν βρήκαμε την απαραίτητη βιβλιογραφία και εφόσον την μελετήσαμε, θέσαμε σαν επόμενο στάδιο να αρχίσουμε να κατανοούμε τον κώδικα που βρήκαμε στην ιστοσελίδα [18] όπου αποφασίσαμε ότι θα αποτελούσε τον αλγοριθμικό σκελετό που θα ακολουθούσαμε. Επομένως θέσαμε τους ακριβείς στόχους της εργασίας και αρχίσαμε να τους υλοποιούμε. Έπειτα κατασκευάσαμε ένα αλγόριθμο που θα αποτελούσε/δημιουργούσε τα μοντέλα RNN και μετά κατασκευάσαμε ένα αλγόριθμο που θα αποτελούσε/δημιουργούσε τα μοντέλα ARIMA με σκοπό να τα συγκρίνουμε και να τα αξιολογήσουμε σύμφωνα με τους στόχους που θέσαμε. Εφόσον είχαμε ολοκληρώσει τον κώδικα μας, τρέξαμε πολλά πειράματα ώστε να πάρουμε τα αποτελέσματα μας και να μπορούμε να αξιολογήσουμε τους δύο αυτούς αλγορίθμους. Έπειτα αξιολογήσαμε πως οι μέθοδοι αυτοί προσφέρουν μια καλύτερη απόδοση εφαρμογής χρησιμοποιώντας την μετρική DRAM hit rate. Στο τέλος αρχίσαμε να συγγράφουμε την συγκεκριμένη διπλωματική εργασία.

Κεφάλαιο 2

Υπόβαθρο Μηχανικής Μάθησης

2.1 Ορισμός	7
2.2 Είδη Μηχανικής Μάθησης	7
2.3 Τεχνητά Νευρωνικά Δίκτυα	9
2.4 Επαναλαμβανόμενα Νευρωνικά Δίκτυα	11
2.5 LSTMs	13
2.6 Συναρτήσεις Ενεργοποίησης	14
2.7 Συνάρτηση Κόστους	15
2.8 Τεχνικές Βελτιστοποίησης και Αλγόριθμοι	16

2.1 Ορισμός

Η μηχανική μάθηση, η οποία ορίστηκε από τον Άρθουρ Σάμουελ το 1959 [16] είναι μια υποδιαίρεση της τεχνητής νοημοσύνης. Χαρακτηρίζεται σε μεγάλο βαθμό από την ικανότητα μιας μηχανής να ενεργεί με τρόπο ανάλογο με τον οποίο ένας άνθρωπος θα έλυνε ένα πρόβλημα. Για τη δημιουργία μοντέλων μηχανικής μάθησης, χρησιμοποιούνται δεδομένα εκπαίδευσης και αυτό επιτρέπει στο μηχανήμα/υπολογιστικό σύστημα να λαμβάνει αποφάσεις χωρίς να είναι προγραμματισμένο να το κάνει. Ένα μέρος της μηχανικής μάθησης σχετίζεται στενά με την υπολογιστική στατιστική, η οποία επικεντρώνεται στη δημιουργία προβλέψεων/προγνώσεων μέσω υπολογιστικών συστημάτων. Η μαθηματική βελτιστοποίηση συνεισφέρει μεθόδους, θεωρίες και τομείς εφαρμογής στον τομέα της μηχανικής μάθησης.

2.2 Είδη Μηχανικής Μάθησης

Η επιλογή του τύπου αλγόριθμου μηχανικής μάθησης που θα χρησιμοποιηθεί εξαρτάται από τα διαθέσιμα δεδομένα. Τα επισημασμένα δεδομένα, τα οποία περιλαμβάνουν παραμέτρους εισόδου και εξόδου αναγνώσιμα από τη μηχανή και τα μη επισημασμένα δεδομένα, που έχουν είτε μία είτε καμία από τις παραμέτρους σε αναγνώσιμη μορφή από μηχανή, είναι οι δύο ποικιλίες δεδομένων που οι αλγόριθμοι μηχανικής μάθησης μπορούν να χρησιμοποιήσουν για να εκπαιδευτούν. Ενώ τα δεδομένα με ετικέτα απαιτούν περισσότερη

προσπάθεια για να αποκτηθούν λόγω σημαντικής ανθρώπινης εργασίας που χρειάζεται για την επισήμανσης τους. Η πολυπλοκότητα των λύσεων που δημιουργούνται με τη χρήση δεδομένων χωρίς ετικέτα μπορεί να είναι μεγαλύτερη.

Επί του παρόντος, τρεις κύριες μέθοδοι μηχανικής μάθησης που χρησιμοποιούνται ευρέως είναι οι εξής:

I. Επιβλεπόμενη Μάθηση (Supervised Learning)

Η επιβλεπόμενη μάθηση είναι ένας θεμελιώδης τύπος μηχανικής μάθησης όπου χρησιμοποιούνται δεδομένα με χαρακτηριστικά (features) αλλά και με ετικέτες (labels) για την εκπαίδευση ενός μοντέλου. Απαιτείται η ακριβής επισήμανση δεδομένων για να είναι αποτελεσματική, κάτι που μπορεί να είναι δύσκολο να επιτευχθεί. Αυτή η προσέγγιση περιλαμβάνει την παροχή ενός μικρού εκπαιδευτικού συνόλου σε έναν αλγόριθμο μηχανικής μάθησης, το οποίο συνήθως αποτελεί μέρος ενός μεγαλύτερου συνόλου δεδομένων και έχει παρόμοια χαρακτηριστικά. Αυτό το σύνολο δεδομένων παρέχει στον αλγόριθμο παραμέτρους που πρέπει να επισημαίνονται με ακρίβεια, προκειμένου να λυθεί σωστά το πρόβλημα. Στη συνέχεια, ο αλγόριθμος επιχειρεί να προσδιορίσει τη σχέση μεταξύ των μεταβλητών εισόδου και εξόδου. Μετά την εκπαίδευση, το μοντέλο μπορεί να εντοπίσει τη σύνδεση μεταξύ τους και να κατανοήσει πώς λειτουργούν τα δεδομένα. Αυτή η μέθοδος χρησιμοποιείται συνήθως για την επίλυση προβλημάτων παλινδρόμησης και ταξινόμησης. Παράδειγμα προβλήματος παλινδρόμησης περιλαμβάνει την πρόβλεψη τιμών μετοχών (χρονολογικές σειρές), ενώ παράδειγμα προβλήματος ταξινόμησης περιλαμβάνει την ανίχνευση είδους ζώου ανά κατηγορίες.

II. Μη επιβλεπόμενη Μάθηση (Unsupervised Learning)

Το κύριο όφελος της μη επιβλεπόμενης μηχανικής μάθησης είναι η ικανότητά της να επεξεργάζεται πληροφορίες από χαρακτηριστικά αλλά χωρίς ετικέτες. Αυτό σημαίνει ότι δεν απαιτείται χειρωνακτική εργασία προκειμένου τα δεδομένα να είναι αναγνώσιμα από το πρόγραμμα. Αυτό συνεπάγεται ότι το σύστημα μπορεί να χειριστεί μεγαλύτερα σύνολα δεδομένων. Αντίθετα, η εποπτευόμενη μάθηση επιτρέπει στον αλγόριθμο να ανιχνεύει την ακριβή σύνδεση μεταξύ δύο δεδομένων σημείων. Ωστόσο, λόγω της έλλειψης ετικετών στη μάθηση χωρίς επίβλεψη, παράγει μια ποικιλία από κρυφές δομές. Το μηχάνημα αναγνωρίζει τη σύνδεση μεταξύ σημείων δεδομένων με πιο αφηρημένο τρόπο χωρίς την ανάγκη για καθορισμένες από τον άνθρωπο παραμέτρους. Αυτό είναι που κάνει τους αλγόριθμους μάθησης

χωρίς επίβλεψη τόσο ευέλικτους. Μπορούν να προσαρμόσουν τις κρυφές δομές ως απάντηση στα δεδομένα, χωρίς να χρειάζονται ένα προκαθορισμένο σύνολο ερωτήσεων ή στόχων. Αυτή η μέθοδος χρησιμοποιείται συνήθως για την επίλυση προβλημάτων ομαδοποίησης, συσχέτισης (clustering) και ανίχνευση ανωμαλιών (anomaly detection). Παράδειγμα προβλήματος ομαδοποίησης είναι όταν προσδιορίζονται ομάδες στα δεδομένα με βάση τα χαρακτηριστικά τους πχ. χρώμα ζώου. Ένα παράδειγμα του προβλήματος συσχέτισης αποτελεί την εύρεση γεγονότων που συμβαίνουν μαζί πχ. όταν ένας άνθρωπος αγοράζει φαγητό παίρνει και ένα ποτό για να το συνοδέψει. Ένα πρόβλημα ανίχνευσης ανωμαλιών μπορεί να είναι ο εντοπισμός φαγητών που διατηρούνται περισσότερο έξω από το ψυγείο.

III. Ενισχυτική Μάθηση (Reinforcement Learning)

Η ενισχυτική μάθηση βασίζεται στην ιδέα του πώς οι άνθρωποι χρησιμοποιούν τα δεδομένα στα οποία εκτίθενται για μάθηση. Χρησιμοποιείται ένας αλγόριθμος που χρησιμοποιεί μια προσέγγιση δοκιμής και λάθους για να βελτιωθεί και να μάθει από νέες καταστάσεις δεδομένων. Οι εκροές που θεωρούνται ωφέλιμες ανταμείβονται, ενώ οι ανεπιτυχείς επιπλήττονται. Σε αυτό το σχήμα, ο αλγόριθμος ενίσχυσης μάθησης τοποθετείται σε ένα περιβάλλον με διερμηνέα και σύστημα ανταμοιβής. Κάθε επανάληψη του αλγορίθμου έχει τον διερμηνέα να αξιολογεί το αποτέλεσμα εξόδου και να καθορίζει εάν είναι ευνοϊκό ή όχι. Εάν το πρόγραμμα ανακαλύψει μια επιτυχημένη λύση, ο διερμηνέας την ενισχύει δίνοντας στον αλγόριθμο μια ανταμοιβή. Σε περίπτωση δυσμενούς έκβασης, ο αλγόριθμος αναγκάζεται να επαναλάβει τη διαδικασία μέχρι να επιτευχθεί καλύτερο αποτέλεσμα. Γενικά, η αποτελεσματικότητα συνδέεται με το σύστημα ανταμοιβής-τιμωρίας.

2.3 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα (ANN) είναι υπολογιστικά συστήματα που βασίζονται σε μεγάλο βαθμό στην αρχιτεκτονική των βιολογικών νευρωνικών δικτύων που βρίσκονται σε εγκεφάλους ζώων. Αυτά τα συστήματα αποτελούνται από διασυνδεδεμένους τεχνητούς νευρώνες, οι οποίοι μιμούνται τις συνάψεις σε έναν βιολογικό εγκέφαλο. Η είσοδος και η έξοδος αυτών των νευρώνων είναι πραγματικοί αριθμοί και η έξοδος τους καθορίζεται από μια μη γραμμική συνάρτηση που παίρνει το άθροισμα των εισόδων ως όρισμα. Αυτοί οι νευρώνες οργανώνονται συνήθως σε στρώματα, με τα σήματα να ταξιδεύουν από το στρώμα εισόδου στο στρώμα εξόδου, αφού περάσουν από αρκετούς μετασχηματισμούς. Αυτό μπορούμε να το δούμε με γραφικό τρόπο στο Σχήμα 2.1.

Για να εκπαιδευτεί ένα νευρωνικό δίκτυο, οι παράμετροι όπως οι μέθοδοι εκπαίδευσης, τα εργαλεία, τα δεδομένα και ο αριθμός των εποχών πρέπει να συντονιστούν προσεκτικά. Αυτό γίνεται για να ελαχιστοποιηθούν οι πόροι και ο χρόνος, καθώς και για να διασφαλιστεί ότι το εκπαιδευμένο δίκτυο είναι σε θέση να γενικεύει και να κάνει ακριβείς προβλέψεις για νέα δεδομένα. Η μαζική εκπαίδευση χρησιμοποιείται για την ενημέρωση των βαρών του δικτύου. Η διαδικασία εκπαίδευσης περιλαμβάνει πολλαπλούς υπολογισμούς διαφορών και παραγώγων και εξαρτάται από παραμέτρους, συμπεριλαμβανομένων των συναρτήσεων ενεργοποίησης, κόστους και βελτιστοποίησης.

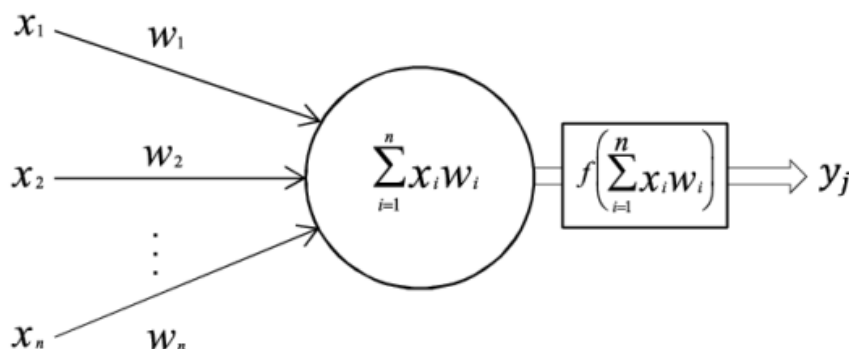
Ας δούμε τα βασικά χαρακτηριστικά των τεχνητών νευρωνικών δικτύων:

- I. Αρχιτεκτονική πολλαπλών επιπέδων: Τα τεχνητά νευρωνικά δίκτυα αποτελούνται από πολλά επίπεδα νευρώνων που συνδέονται μεταξύ τους. Κάθε επίπεδο λαμβάνει εισόδους από το προηγούμενο επίπεδο, επεξεργάζεται τις πληροφορίες και παράγει την έξοδο που μεταδίδεται στο επόμενο επίπεδο. Αυτή η πολύ επίπεδη δομή επιτρέπει την αναπαράσταση και εξαγωγή πολύπλοκων χαρακτηριστικών από τα δεδομένα εισόδου.
- II. Τεχνητοί νευρώνες: Οι τεχνητοί νευρώνες λειτουργούν ως μαθηματικά μοντέλα που προσομοιώνουν τη λειτουργία των βιολογικών νευρώνων. Λαμβάνουν εισόδους από άλλους νευρώνες ή από τα δεδομένα εισόδου και εφαρμόζουν μια συνάρτηση ενεργοποίησης για να υπολογίσουν την έξοδό τους. Η συνάρτηση ενεργοποίησης μπορεί να είναι συνήθως μια μη γραμμική συνάρτηση, όπως η σιγμοειδής συνάρτηση.
- III. Συνδέσεις μεταξύ νευρώνων: Οι νευρώνες συνδέονται μεταξύ τους με βάρη που καθορίζουν τη δύναμη και την κατεύθυνση της επικοινωνίας. Κάθε σύνδεση έχει ένα συγκεκριμένο βάρος που αντιστοιχεί στην επίδραση που έχει ο ένας νευρώνας στον άλλο. Κατά την εκπαίδευση, τα βάρη προσαρμόζονται ώστε το νευρωνικό δίκτυο να μπορεί να μάθει και να προσαρμόζει τη συμπεριφορά του.
- IV. Εκπαίδευση: Τα τεχνητά νευρωνικά δίκτυα εκπαιδεύονται μέσω αλγόριθμου εκπαίδευσης. Ο αλγόριθμος υπολογίζει το σφάλμα μεταξύ της εξόδου του νευρωνικού δικτύου και της επιθυμητής εξόδου, και χρησιμοποιεί αυτό το σφάλμα για να ενημερώνει τα βάρη των νευρώνων, προσαρμόζοντάς τα ώστε να μειώσει το σφάλμα στην επόμενη εκπαιδευτική επανάληψη (εποχή). Ο δημοφιλέστερος αλγόριθμος εκπαίδευσης των τεχνητών νευρωνικών δικτύων είναι ο αλγόριθμος προς

τα πίσω [6] (backpropagation) ο οποίος χρησιμοποιεί τη μέθοδο της κατάβασης κλίσης (gradient descent) για την ενημέρωση των βαρών.

Τα τεχνητά νευρωνικά δίκτυα έχουν τη δυνατότητα να εξάγουν συμπεράσματα από μεγάλες ποσότητες δεδομένων καθώς και να αντιλαμβάνονται πολύπλοκα μοτίβα και δομές. Αυτό τους επιτρέπει να εκτελούν προβλέψεις και να λαμβάνουν αποφάσεις με βάση τα προηγούμενα παραδείγματα και την εμπειρία που έχουν αποκτήσει κατά τη διάρκεια της εκπαίδευσης.

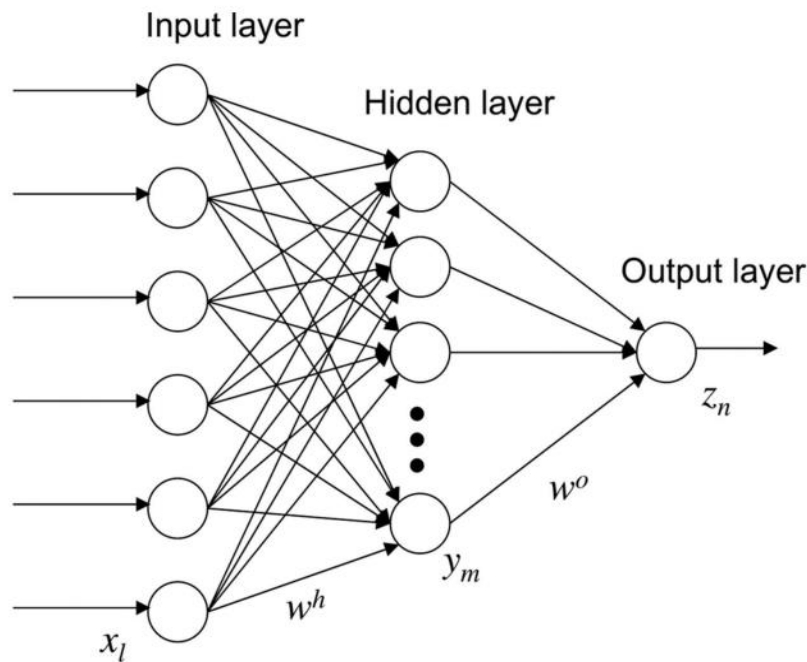
Ένα από τα κύρια πλεονεκτήματα των τεχνητών νευρωνικών δικτύων είναι η δυνατότητά τους να προσαρμόζονται και να μαθαίνουν από τα δεδομένα, χωρίς να χρειάζεται να προγραμματιστούν ακριβείς κανόνες. Αυτό τα καθιστά ευέλικτα και ικανά να αντιμετωπίσουν προβλήματα για τα οποία η ακριβής περιγραφή ή το ακριβές μοντέλο είναι δύσκολο να διατυπωθεί. Ωστόσο, αυτή η ευελιξία μπορεί να οδηγήσει σε έναν αυξημένο αριθμό παραμέτρων και πιθανών συνδυασμών, απαιτώντας μεγάλο όγκο δεδομένων και υπολογιστική ισχύ για την εκπαίδευση και την αξιολόγησή.



Σχήμα 2.1 Ένας νευρώνας που δέχεται συγκεκριμένα βάρη και αυξομειώνει την έξοδο του.

2.4 Επαναλαμβανόμενα Νευρωνικά Δίκτυα

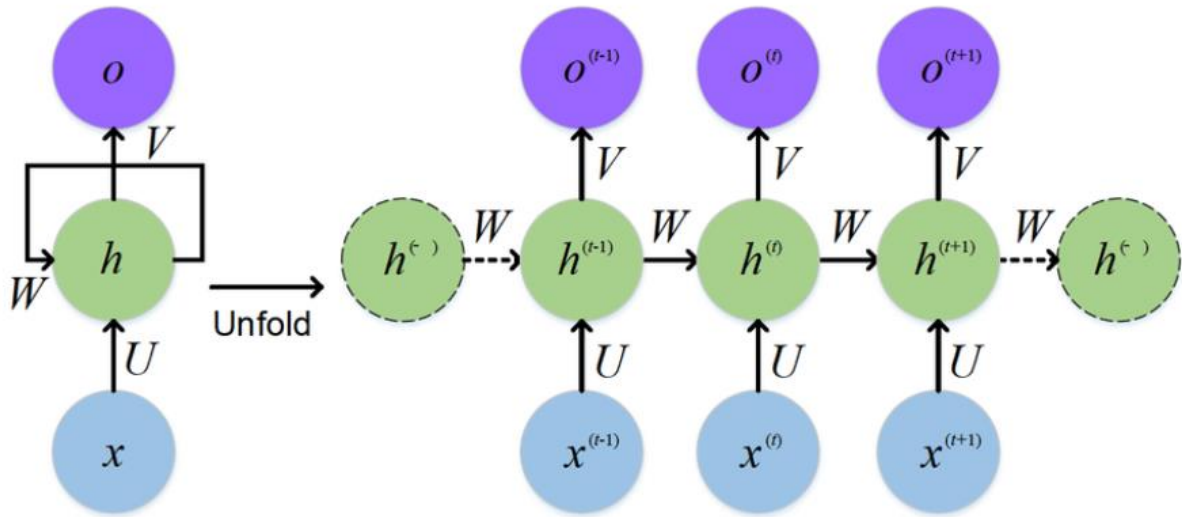
Πολλά νευρωνικά δίκτυα χρησιμοποιούν διάταξη τροφοδοσίας. Σε αυτή τη διάταξη, τα δεδομένα διακινούνται σε μονή κατεύθυνση. Συνήθως, αυτή η μεταφορά δεδομένων ξεκινά από μια ομάδα κόμβων εισόδου που συνδέονται με ένα σύνολο κρυφών κόμβων. Στη συνέχεια, αυτοί οι κρυφοί κόμβοι συνδέονται με ένα σύνολο κόμβων εξόδου. Μια απεικόνιση αυτής της δομής για καλύτερη κατανόηση παρουσιάζεται στο Σχήμα 2.2.



Σχήμα 2.2 Ένα νευρωνικό δίκτυο με τα δεδομένα να μεταφέρονται μόνο προς μια κατεύθυνση από το επίπεδο εισόδου στο κρυφό επίπεδο και έπειτα στο επίπεδο εξόδου.

Τα δίκτυα προώθησης τροφοδοσίας χρησιμοποιούνται συχνά για εργασίες όπου η σειρά των δεδομένων εισόδου δεν είναι απαραίτητη. Χαρακτηριστικό παράδειγμα αυτού είναι η ταξινόμηση εικόνων, καθώς δεν υπάρχει χρονική συσχέτιση μεταξύ των μεμονωμένων στοιχείων στη βάση δεδομένων εικόνων. Επομένως, τα δίκτυα κατηγοριοποίησης εικόνας τείνουν να τυχαioποιούν τη σειρά των δεδομένων εισόδου σε κάθε εποχή εκπαίδευσης για να αποφευχθεί το συμπέρασμα μιας ανύπαρκτης σχέσης.

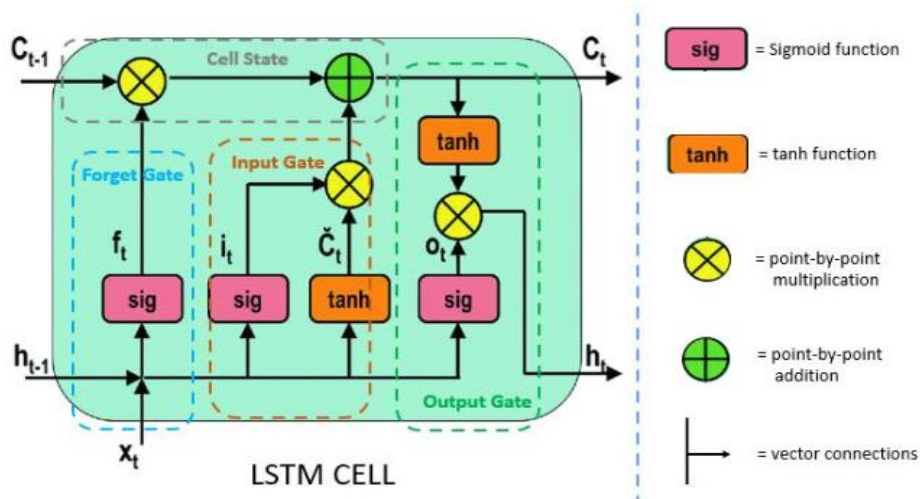
Αντίθετα, τα επαναλαμβανόμενα νευρωνικά δίκτυα χρησιμοποιούνται όταν η χρονική συσχέτιση μεταξύ διαφορετικών στοιχείων σε ένα σύνολο δεδομένων είναι σχετική. Οι κόμβοι στο κρυφό στρώμα αυτών των δικτύων λαμβάνουν πληροφορίες τόσο από τους κόμβους εισόδου όσο και από τους ίδιους. Με αυτόν τον τρόπο, τα δεδομένα από το παρελθόν αποθηκεύονται και χρησιμοποιούνται για να γίνουν προσδιορισμοί με βάση τα συμφοραζόμενα σχετικά με το τι φαίνεται στο παρόν. Αυτά τα δεδομένα μπορούν να κατανεμηθούν σε πολλά χρονικά βήματα και η απεικόνιση του τρόπου με τον οποίο μεταδίδονται οι πληροφορίες από το ένα χρονικό βήμα στο επόμενο αναφέρεται ως ξεδίπλωμα RNN (unfold), όπως φαίνεται στο σχήμα 2.3.



Σχήμα 2.3 Ξεδίπλωμα ενός απλού νευρωνικού δικτύου κατά τι διάρκεια του χρόνου. (Το t συμβολίζει τον χρόνο, το o συμβολίζει την έξοδος και το x συμβολίζει την είσοδο του νευρώνα) [13].

2.5 LSTMs

Το 1997, οι Hochreiter και Schmidhuber [5] πρότειναν τη Μακροπρόθεσμη Βραχυπρόθεσμη Μνήμη (LSTM) ως τεχνική για την αντιμετώπιση του ζητήματος της φθίνουσας κλίσης των επαναλαμβανόμενων νευρωνικών δικτύων. Ως εξελιγμένη έκδοση του πρώτου, το LSTM χρησιμοποιείται ευρέως στα σημερινά δίκτυα. Η μνήμη του είναι γνωστή ως κατάσταση κυψέλης C , η οποία ρυθμίζεται από μια σειρά από πύλες: την πύλη λήθης F , την πύλη εισόδου I και την πύλη εξόδου O . Οι οποίες είναι Νευρωνικά Δίκτυα που τροποποιούνται κατά τη διάρκεια της εκπαίδευσης. Όπως απεικονίζεται στο Σχήμα 2.4, η πύλη εισόδου I αποφασίζει ποιες νέες πληροφορίες θα προσθέσει στη κυψέλη C , η πύλη λήθης F καθορίζει ποιες πληροφορίες θα απορριφθούν και η πύλη εξόδου O επιλέγει ποιες πληροφορίες θα εξάγει από το κελί LSTM σε κάθε χρονικό βήμα.



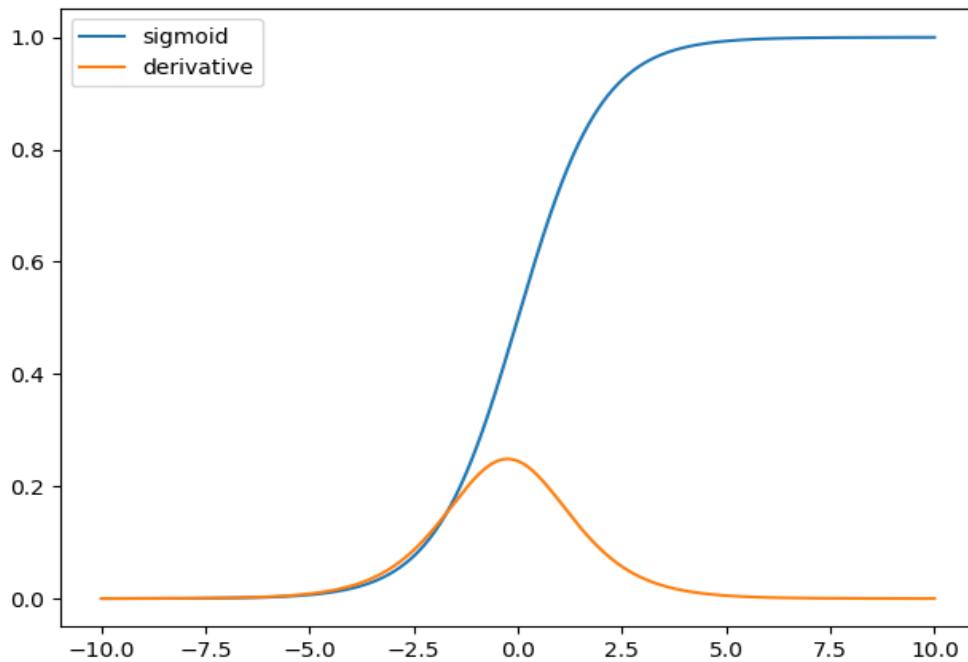
Σχήμα 2.4 Ένα LSTM CELL [12]

Το πιο ουσιαστικό στοιχείο των LSTM είναι η κατάσταση κυψέλης C , η οποία επιτρέπει τη ροή πληροφοριών μεταξύ πολλαπλών χρονικών βημάτων έως και εκατομμυρίων κυψελών LSTM. Το πρόβλημα της εξαφάνισης κλίσης δεν αποτελεί πλέον πρόβλημα, καθώς η πύλη εισόδου I και η πύλη λήθης F ελέγχουν την προσθήκη και την αφαίρεση πληροφοριών με την πάροδο του χρόνου, επιτρέποντας τη μετάδοση των υπολογισμένων σφαλμάτων πίσω στο χρόνο. Αυτό έχει κάνει τα LSTM δημοφιλή σε πολλούς τομείς, ειδικά σε εφαρμογές Επεξεργασίας Φυσικής Γλώσσας (NLP), όπως η αναγνώριση ομιλίας και η μετάφραση γλώσσας κειμένου σε κείμενο, όπου χρησιμοποιούνται μεγάλες ακολουθίες δεδομένων [7].

2.6 Συναρτήσεις Ενεργοποίησης

Η σημασία των λειτουργιών ενεργοποίησης στη βαθιά μάθηση (Deep Learning) πρέπει να υπερτονιστεί καθώς είναι υπεύθυνη για τον προσδιορισμό της εξόδου κάθε κόμβου και κατά συνέπεια της εξόδου του συνολικού συστήματος. Είναι επομένως καθοριστικός παράγοντας σε πολλές πτυχές του μοντέλου, όπως η ταχύτητα εκπαίδευσης, η ικανότητα σύγκλισης του δικτύου και η ακρίβεια των προβλέψεών του. Για λόγους πληρότητας, παρέχουμε μια σύντομη επισκόπηση δύο χρήσιμων συναρτήσεων ενεργοποίησης:

- Sigmoid Function: Η συνάρτηση Sigmoid είναι μια συνάρτηση ενεργοποίησης που χρησιμοποιήθηκε για πρώτη φορά στην εκπαίδευση των νευρωνικών δικτύων. Το πεδίο των εισόδων της είναι το σύνολο όλων των πραγματικών αριθμών, ενώ το εύρος της είναι $(0,1)$. Ακόμη και όταν η είσοδος είναι ένας πολύ μεγάλος θετικός ή αρνητικός αριθμός, η έξοδος παραμένει μεταξύ 0 και 1. Αυτό έχει ως αποτέλεσμα μικρές παραγώγους και κατά συνέπεια μια πιο αργή διαδικασία μάθησης γνωστή ως Vanishing Gradient, η οποία είναι υπεύθυνη για πολλά προβλήματα στη μαθησιακή διαδικασία.



Σχήμα 2.5 Σιγμοειδή συνάρτηση. Η έξοδος της σιγμοειδούς συνάρτησης μπορεί να προσδιοριστεί μαθηματικά χρησιμοποιώντας την ακόλουθη εξίσωση $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$.

- Η Softmax είναι μια γενικευμένη μορφή της σιγμοειδούς συνάρτησης, που χρησιμοποιείται συχνά σε προβλήματα ταξινόμησης πολλών κλάσεων. Κανονικοποιεί τις τιμές εξόδου τοποθετώντας τις στο εύρος του (0,1), με την προστιθέμενη ιδιότητα ότι τα συστατικά αθροίζονται στο 1 καθιστώντας τα ερμηνεύσιμα ως πιθανότητες. Η έξοδος της συνάρτησης Softmax εκφράζεται μαθηματικά ως εξής:

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

2.7 Συνάρτηση Κόστους

Η συνάρτηση κόστους χρησιμοποιείται ως εργαλείο παρακολούθησης για την επαναληπτική διαδικασία της εκπαίδευσης. Παρέχει μια αξιολόγηση του πόσο αναποτελεσματικό είναι ένα μοντέλο πρόβλεψης. Με απλούς όρους, μια συνάρτηση κόστους είναι ένα μέτρο του πόσο ανακριβές είναι το μοντέλο όσον αφορά την ικανότητά του να υπολογίζει τη σχέση μεταξύ y και x . Συνήθως, αυτό απεικονίζεται ως ένα χάσμα μεταξύ της πραγματικής και της προβλεπόμενης τιμής. Η συνάρτηση κόστους υπολογίζεται όταν τρέχουμε το μοντέλο μας κατά την αξιολόγηση των εκτιμώμενων αποτελεσμάτων έναντι των προβλεπόμενων τιμών για το y .

Μία από τις πιο γνωστές συναρτήσεις κόστους είναι η Απώλεια Διασταυρούμενης Εντροπίας (Cross-Entropy Loss) η οποία αναπαρίσταται μαθηματικά ως εξής:

$$J(\theta) = -H(y, p) = -\sum_{i=0}^N \hat{y}_i \log(p_i, j)$$

Έστω N ο αριθμός των διακριτών κλάσεων και έστω $\mathcal{Y}_i$ η εκτιμώμενη τιμή για την παρατήρηση i και $p_{i,j} = p(\hat{y}_i = j|x)$ η οπίσθια πιθανότητα i να ανήκει στην κλάση j . Η εξίσωση που δίνεται παραπάνω υπολογίζει το κόστος κάθε δείγματος εισόδου και για να λάβουμε το συνολικό κόστος υπολογίζουμε απλώς τον μέσο όρο όλων των επιμέρους δαπανών.

Μια άλλη διαδομένη συνάρτηση κόστους είναι το μέσο τετραγωνικό σφάλμα (MSE) που χρησιμοποιείται συνήθως για να μετρά την εγγύτητα μιας γραμμής παλινδρόμησης σε ένα σύνολο σημείων. Αυτό γίνεται με τον υπολογισμό των αποστάσεων μεταξύ των σημείων και της γραμμής παλινδρόμησης, οι οποίες αναφέρονται ως «σφάλματα». Αυτά τα σφάλματα τετραγωνίζονται για να αφαιρέσουν τυχόν αρνητικά σημάδια, καθώς και για να τονίσουν μεγαλύτερες αποκλίσεις. Ο όρος "μέσο τετραγωνικό σφάλμα" προέρχεται από το γεγονός ότι είναι ο μέσος όρος ενός συνόλου σφαλμάτων. Όσο χαμηλότερο είναι το MSE, τόσο πιο ακριβής είναι η πρόβλεψη. Η μαθηματική αναπαράσταση της είναι η εξής:

$$J(\theta) = \frac{1}{n} \sum_{i=0}^M (Y_i - \hat{y}_i)^2$$

2.8 Τεχνικές Βελτιστοποίησης και Αλγόριθμοι

Η βελτιστοποίηση αναφέρεται στην πρακτική ελαχιστοποίησης ή μεγιστοποίησης μιας μαθηματικής έκφρασης. Οι βελτιστοποιητές είναι αλγόριθμοι και μέθοδοι που χρησιμοποιούνται στα νευρωνικά δίκτυα για την αλλαγή χαρακτηριστικών, όπως τα βάρη και τον ρυθμό εκμάθησης, προκειμένου να μειωθούν οι απώλειες. Οι βελτιστοποιητές χρησιμοποιούνται για την επίλυση προβλημάτων βελτιστοποίησης ελαχιστοποιώντας ή μεγιστοποιώντας μια συνάρτηση. Τα τελευταία χρόνια, έχει μελετηθεί μια ποικιλία βελτιστοποιητών, οι οποίοι έχουν όλα τα δικά τους πλεονεκτήματα και μειονεκτήματα.

Η Backpropagation είναι η πιο ευρέως χρησιμοποιούμενη τεχνική για την εκπαίδευση τεχνητών νευρωνικών δικτύων. Είναι μια ομάδα διαδικασιών βελτιστοποίησης που μοιράζονται μια επαναλαμβανόμενη και αναδρομική προσέγγιση για τον υπολογισμό των ενημερωμένων βαρών του μοντέλου δικτύου. Πρακτικά, μια λειτουργία Feed-Forward

διεξάγεται στο δίκτυο, παρέχοντάς του μια είσοδο (x) και δημιουργώντας μια έξοδο (y). Στη συνέχεια ξεκινά η διαδικασία Backpropagation για να υπολογιστεί η κλίση της συνάρτησης απώλειας, σε σχέση με το βάρος κάθε νευρώνα $\frac{\partial J(\theta)}{\partial \omega_k}$. Ο Κανόνας της Αλυσίδας χρησιμοποιείται εδώ, υπολογίζοντας τη διαβάθμιση στρώμα προς στρώμα σε κατεύθυνση προς τα πίσω για να αποφευχθεί η επανάληψη τυχόν ενδιάμεσων βημάτων. Η δημοτικότητα αυτής της ομάδας μεθόδων βελτιστοποίησης οφείλεται κυρίως στην αποτελεσματικότητά τους στον υπολογισμό των τιμών βάρους, καθώς και στην ικανότητά τους να αναπροσαρμόζουν γρήγορα τυχόν αυθαίρετες τιμές μεροληψίας νευρώνων.

Το Gradient descent είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την εύρεση του τοπικού ελάχιστου μιας συνάρτησης κόστους. Οι παράμετροι του προβλήματος τροποποιούνται επαναληπτικά σύμφωνα με τις κλίσεις της συνάρτησης κόστους μέχρι να επιτευχθεί σύγκλιση ή να επιτευχθεί ένας προκαθορισμένος αριθμός επαναλήψεων (κριτήρια τερματισμού). Οι παράμετροι του δικτύου τροποποιούνται χρησιμοποιώντας την ακόλουθη εξίσωση: $\theta_{t+1} = \theta_t - \lambda \nabla_{\theta} J(\theta)$. Ο αριθμός των παραμέτρων του δικτύου συμβολίζεται με θ , ο ρυθμός εκμάθησης συμβολίζεται με λ και η συνάρτηση κόστους εκφράζεται ως $J(\theta)$. Γενικά, χρησιμοποιούνται αλγόριθμοι Gradient Descent, αν και η συγκεκριμένη προσέγγιση εξαρτάται από τον όγκο των δεδομένων προς επεξεργασία.

Το Batch Gradient Descent, γνωστό και ως vanilla Gradient Descent, περιλαμβάνει τον υπολογισμό του σφάλματος για κάθε παράδειγμα στο σύνολο δεδομένων πριν από την ενημέρωση του μοντέλου. Αυτός ο κύκλος αναφέρεται ως εποχή εκπαίδευσης. Συγκριτικά, το Stochastic Gradient Descent (SGD) ενημερώνει τις παραμέτρους για κάθε παράδειγμα εκπαίδευσης ξεχωριστά, καθιστώντας το ταχύτερο από το Batch Gradient Descent σε ορισμένες περιπτώσεις. Επιπλέον, οι συχνές ενημερώσεις που παρέχονται από την SGD επιτρέπουν έναν πιο ακριβή ρυθμό βελτίωσης. Η mini-batch Gradient Descent είναι η προτιμώμενη μέθοδος, καθώς επιτυγχάνει μια ισορροπία μεταξύ των δύο χωρίζοντας το σύνολο δεδομένων εκπαίδευσης σε παρτίδες και ενημερώνοντας το μοντέλο για κάθε παρτίδα.

Δεν υπάρχει αμφιβολία ότι ο αλγόριθμος Gradient Descent έχει ορισμένα μειονεκτήματα, προκαλώντας περαιτέρω τροποποιήσεις της κλασικής έκδοσης. Ο σταθερός ρυθμός εκμάθησής του και η αργή σύγκλιση σε προβλήματα με μεγάλα σύνολα δεδομένων (λόγω της μεγάλης διακύμανσης της κλίσης) καθιστούν αναγκαία τη δημιουργία αλγορίθμων προσαρμοσμένων σε αυτά τα συγκεκριμένα προβλήματα.

Ο Adam optimizer (Adaptive Moment Estimations) βασίζεται στο SGD, αλλά χρησιμοποιεί ξεχωριστούς ρυθμούς εκμάθησης για κάθε παράμετρο-βάρος του δικτύου. Χρησιμοποιεί τόσο την πρώτη όσο και τη δεύτερη στιγμή κλίσης καθώς και δύο μεταβλητές Forget για να αποτρέψει την ταλάντωση για να επιταχύνει τη σύγκλιση.

Μια συγκρίσιμη προσέγγιση στο ποσοστό μάθησης υιοθετείται από τον αλγόριθμο Adagrad. Αυτός ο αλγόριθμος μειώνει το ρυθμό εκμάθησης πιο γρήγορα για συχνές παραμέτρους και πιο αργά για σπάνιες παραμέτρους. Ωστόσο, σε ορισμένες περιπτώσεις προκύπτει ένα σημαντικό ζήτημα, όπου ο αποτελεσματικός ρυθμός μάθησης μειώνεται δραματικά λόγω της συσσώρευσης κλίσεων από την αρχή της εκπαίδευσης, με αποτέλεσμα το μοντέλο να φτάσει σε ένα σημείο στο οποίο η διαδικασία μάθησης σχεδόν σταματά λόγω του μικροσκοπικού ποσοστού μάθησης. Τέτοια ζητήματα έχουν αντιμετωπιστεί με αλγόριθμους που δεν απαιτούν αρχικές ρυθμίσεις ρυθμού εκμάθησης και δεν είναι ευαίσθητοι σε υπερπαραμέτρους (πχ. αλγόριθμος Adadelta).

Κεφάλαιο 3

Δρομολογητής Σελίδων

3.1 Εισαγωγή	19
3.2 Επισκόπηση Λειτουργίας Δρομολογητή Σελίδων	20
3.2.1 Ενέργεια πρώτη – Επιλογέας Σελίδας	20
3.2.2 Ενέργεια δεύτερη – Έξυπνος τρόπος διαχείρισης τοποθέτησης σελίδων	22
3.2.3 Ενέργεια τρίτη – Εκχώρηση σελίδων στις μονάδες μνήμης	23
3.3 Χρήση πολιτικής αδειάσματος DRAM από τον δρομολογητή	23

3.1 Εισαγωγή

Έχουμε ακολουθήσει την δημιουργία ενός δρομολογητή παρόμοιου με του KLEIO [3] ο οποίος είναι προσαρμοσμένος για υβριδικά συστήματα μνήμης, ώστε να διαχειρίζεται/δρομολογεί σελίδες εφαρμογών. Δηλαδή να τις τοποθετεί/αφαιρεί από την DRAM και την NVM αναλόγως ορισμένων αποφάσεων κάθε εποχική περίοδο προγραμματισμού (εποχή δρομολόγησης). Μια εποχική περίοδος προγραμματισμού ξεκινά όταν ολοκληρώνονται x αιτήματα της εφαρμογής για πρόσβαση σε σελίδες. Ο δρομολογητής σελίδων βελτιώνει περαιτέρω την απόδοση της εφαρμογής (DRAM hit rate) χρησιμοποιώντας αποφάσεις τοποθέτησης σελίδων εφαρμογών που βασίζονται σε έξυπνες μεθόδους όπως μηχανική μάθηση (LSTM) ή στατιστικά μοντέλα (ARIMA) για μια επιλεγμένη ομάδα σελίδων. Αυτές τις δύο έξυπνες μεθόδους θα τις υλοποιήσουμε και θα τις συγκρίνουμε στο τελευταίο κεφάλαιο της διπλωματικής εργασίας.

Σύμφωνα με τα άρθρα [2,3] η διερεύνηση των ιχνών εφαρμογών (workloads) που περιέχουν όλες τις προσβάσεις σελίδων της κάθε εφαρμογής, προσφέρει πολύτιμες πληροφορίες για παράγοντες που επηρεάζουν την απόδοση μιας εφαρμογής. Οι ερευνητές [2,3,4] έχουν συμπεράνει ότι δεν χρειάζεται έξυπνη διαχείριση σελίδων για όλο το φόρτο εργασίας αλλά μόνο για ένα μικρό μέρος του φόρτου εργασίας. Όπου αν διαχειριστεί έξυπνα τότε θα υπάρχει σημαντική αύξηση της απόδοσης (DRAM hit rate) εφαρμογής. Η έξυπνη αυτή διαχείριση θα γίνει χρησιμοποιώντας ένα μοντέλο επαναλαμβανόμενου νευρωνικού δικτύου

(RNN) ή με ένα μοντέλο ARIMA για κάθε μια από τις πιο ωφέλιμες σελίδες του κάθε φόρτου εργασίας όπου θα κάνει ακριβής προβλέψεις.

Συμπερασματικά ουσιαστική βελτίωση της απόδοσης μπορεί να επιτευχθεί χρησιμοποιώντας ένα έξυπνο τρόπο πρόβλεψης (LSTM & ARIMA) μόνο για ένα μικρό υποσύνολο σελίδων.

3.2 Επισκόπηση Λειτουργίας Δρομολογητή Συστήματος Υβριδικής Μνήμης

Ο προτεινόμενος δρομολογητής σελίδων εκτελείται σε κάθε εποχική περίοδο προγραμματισμού και πρέπει να εκτελεί συγκεκριμένες ενέργειες:

3.2.1 Ενέργεια πρώτη – Επιλογέας Σελίδας

Όπως αναφέρθηκε προηγουμένως, δεν έχουν όλες οι σελίδες ουσιαστικό αντίκτυπο στην απόδοση μιας εφαρμογής. Πολλές σελίδες μπορούν να διαχειρίζονται αποτελεσματικά χρησιμοποιώντας μια πολιτική που βασίζεται στο ιστορικό, εξαλείφοντας την ανάγκη για έξυπνη διαχείριση σελίδων. Ωστόσο, ορισμένες σελίδες απαιτούν πιο έξυπνη διαχείριση. Οπότε ο Επιλογέας Σελίδων είναι υπεύθυνος για το διαχωρισμό των σελίδων που χρειάζονται έξυπνη διαχείριση και αυτών που δεν χρειάζονται. Σε ένα ιδανικό σενάριο με απεριόριστους υπολογιστικούς πόρους, η διαχείριση κάθε σελίδας θα γινόταν χρησιμοποιώντας έξυπνους τρόπους όπως μηχανική μάθηση ή στατιστικά μοντέλα, αλλά αυτό δεν είναι πρακτικό και δεν μπορεί να κλιμακωθεί για εφαρμογές με εκατομμύρια σελίδες.

Η πρόβλεψη που βασίζεται στην πολιτική ιστορικού, που αναφέρεται συχνά σε αυτή τη διπλωματική εργασία, βασίζεται αποκλειστικά σε πληροφορίες από τη προηγούμενη εποχική περίοδο προγραμματισμού σελίδας για να κάνει προβλέψεις για την επόμενη.

Ο Επιλογέας σελίδας χρησιμοποιεί μοντέλα-πολιτικές για την εκτίμηση της απόδοσης της εφαρμογής (DRAM hit rate) προσομοιώνοντας τη διαδικασία εποχικής περιόδου προγραμματισμού σελίδας, δεδομένης μιας συγκεκριμένης διαμόρφωσης συστήματος υβριδικής μνήμης.

Πρώτον, δημιουργεί μια εκτίμηση του χρόνου εκτέλεσης εφαρμογής χρησιμοποιώντας τη πολιτική ιστορικού. Αυτό επιτρέπει την καταγραφή πληροφοριών σχετικά με την εσφαλμένη τοποθέτηση κάθε σελίδας λόγω ανακριβούς εκτίμησης προσβάσεων σελίδας για τη

συγκεκριμένη εποχική περιόδου προγραμματισμού σελίδας όταν χρησιμοποιούνται μόνο ιστορικά δεδομένα.

Ο Επιλογέας σελίδας χωρίζει τις σελίδες σε δύο υποσύνολα ανάλογα με τον αντίκτυπό τους στην απόδοση. Για να κατηγοριοποιήσει τις σελίδες, ο Επιλογέας Σελίδας λαμβάνει υπόψη δύο μετρήσεις για κάθε σελίδα:

1. Τον αριθμός των φορών που έχει γίνει πρόσβαση σε μια σελίδα. Οι σελίδες με συχνή πρόσβαση έχουν τον πιο σημαντικό αντίκτυπο στην απόδοση, επομένως η ιεράρχηση της διαχείρισής τους είναι ζωτικής σημασίας. Αυτό υπολογίζεται με την πιο κάτω πράξη:

$$Accesses_i(x) = 3 * Writes_i(x) + Reads_i(x)$$

Η $Accesses_i(x)$ αντιπροσωπεύει τον αριθμό των φορών που έγινε πρόσβαση στη σελίδα x κατά τη διάρκεια της i-στης εποχικής περιόδου προγραμματισμού.

2. Την ικανότητα του δρομολογητή σελίδων βάσει της πολιτικής ιστορικού να τοποθετεί τη σελίδα στη σωστή μονάδα αποθήκευσης. Εάν η πολιτική ιστορικού μπορεί να τοποθετήσει σωστά τη σελίδα σε διαφορετικά στοιχεία μνήμης, δεν χρειάζεται να εφαρμοστούν τεχνικές μηχανικής μάθησης ή στατιστικά μοντέλα που απαιτούν υπολογιστικούς πόρους για αυτήν τη σελίδα, καθώς δεν θα υπήρχε επιπλέον όφελος από μια πιο ακριβή πρόβλεψη. Ωστόσο, αν ο δρομολογητής σελίδων που βασίζεται στην πολιτικής ιστορικού τοποθετήσει λανθασμένα σελίδες στις μονάδες μνήμης τότε αυτές οι σελίδες δεν είναι ασήμαντες και ένα RNN ή ένα μοντέλο ARIMA θα πρέπει να διαχειρίζεται την τοποθέτηση της κάθε συγκεκριμένης σελίδας. Το πως ορίζουμε αν μια τοποθέτηση είναι λανθασμένη είναι συγκρίνοντας την τοποθέτηση που έγινε με την πολιτική ιστορικού μαζί με την βέλτιστη τοποθέτηση σύμφωνα με την πολιτική oracle.

$$Misplacement_i(x) = \begin{cases} 1 & \text{If Page } x \text{ was misplaced during Period } i \\ 0 & \text{If Page } x \text{ was properly placed during Period } i \end{cases}$$

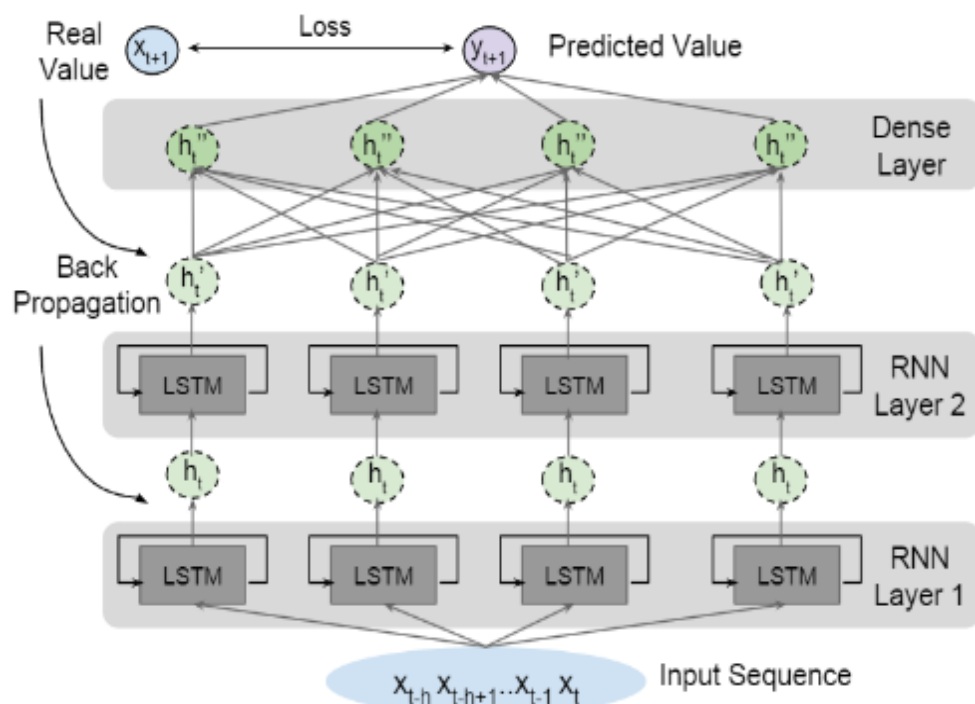
Αυτές οι δύο μετρήσεις συνδυάζονται στον ακόλουθο τύπο για κάθε Σελίδα x σε περιόδους προγραμματισμού i=0...N:

$$Profit(x) = \sum_{i=0}^N Accesses_i(x) * Misplacement_i(x)$$

3.2.2 Ενέργεια δεύτερη – Έξυπνος τρόπος διαχείρισης τοποθέτησης σελίδων

Δύο υποσύνολα σελίδων είναι πλέον διαθέσιμα, το πρώτο περιλαμβάνει κρίσιμες σελίδες που απαιτούν ειδικό χειρισμό και το δεύτερο περιέχει τις υπόλοιπες σελίδες.

Για κάθε κρίσιμη σελίδα, χρησιμοποιείται ένα ξεχωριστό μοντέλο RNN ή ARIMA. Η ύπαρξη ενός μοναδικού μοντέλου ανά σελίδα, όταν ο συνολικός αριθμός σελίδων μπορεί να φτάσει τις εκατοντάδες χιλιάδες, είναι συγκρίσιμος με την ύπαρξη ενός μόνο μοντέλου RNN ή ARIMA που κάνει προβλέψεις σε αυτές τις σελίδες, καθώς το μέγεθος του προβλήματος εισόδου παραμένει σταθερό. Τα μοντέλα μας χρησιμοποιούν δίκτυα που βασίζονται σε κελιά LSTM, με δύο στοιβαγμένα επίπεδα LSTM που ακολουθούνται από ένα επίπεδο εξόδου (dense layer).



Το σχήμα 3.1 παρουσιάζει μια απλοποιημένη επισκόπηση διάταξης ενός RNN δικτύου [3].

Μια προσέγγιση βασισμένη στην πολιτική ιστορικού χρησιμοποιείται για τις άλλες μη κρίσιμες σελίδες, υποθέτοντας ότι θα διατηρήσουν τον αριθμό πρόσβασής τους στην επόμενη εποχή.

3.2.3 Ενέργεια τρίτη – Εκχώρηση σελίδων στις μονάδες μνήμης

Έχοντας συγκεντρωθεί οι μετρήσεις πρόσβασης ανά σελίδα για την επόμενη περίοδο προγραμματισμού, οι σελίδες ταξινομούνται με φθίνουσα σειρά της προβλεπόμενης συχνότητας πρόσβασης. Οι σελίδες με τη συχνότερη πρόσβαση εκχωρούνται στη μνήμη DRAM μέχρι να επιτευχθεί η χωρητικότητα, ενώ οι υπόλοιπες σελίδες τοποθετούνται στη πιο αργή NVM.

3.3 Χρήση πολιτικής αδειάσματος DRAM από τον δρομολογητή

Μια κρίσιμη πτυχή ενός Δρομολογητή Σελιδών που δεν έχει συζητηθεί ακόμη είναι η πολιτική αδειάσματος σελίδων από την DRAM, η οποία απαιτεί μια μέθοδο για να αποφασίσει ποιο στοιχείο-σελίδα θα αφαιρεθεί από την μνήμη.

Όπως αναφέραμε και στις πιο πάνω ενότητες ο δρομολογητής σελίδων συγκεντρώνει τις συχνότητες πρόσβασης ανά σελίδα για κάθε εποχή δρομολόγησης με βάση το ίχνος πρόσβασης εφαρμογής (workload), εκτελώντας στη συνέχεια μεταναστεύσεις μεταξύ γρήγορης (DRAM) και αργής μνήμης (NVM).

Ο δρομολογητής σελίδων σε κάθε εποχή δρομολόγησης, διακρίνει τις σελίδες στις οποίες έχει προσπελαστεί συχνά (δηλαδή τις σελίδες που θεωρούνται ωφέλιμες) και μετατοπίζει οποιεσδήποτε τέτοιες σελίδες βρίσκονται στην NVM στην DRAM. Αυτή η μετατόπιση σελίδων γίνεται με μια ευρέως κοινή χρησιμοποιούμενη πολιτική αδειάσματος την LRU (Last Recently Used). Η οποία επίσης είναι δημοφιλής ως πολιτική αδειάσματος προσωρινής μνήμης.

Ο αριθμός των μετεγκαταστάσεων που πραγματοποιούνται μέσα σε μια δεδομένη περίοδο δρομολόγησης σελίδων δεσμεύεται από τη διαθέσιμη χωρητικότητα της DRAM, καθώς η διαδικασία ανταλλαγής περιλαμβάνει την εναλλαγή ωφέλιμων σελίδων και LRU σε όλη την υβριδική μνήμη. Αυτή η διαδικασία εναλλαγής εμφανίζεται ασύγχρονα και διαδοχικά σύμφωνα με τη σειρά των ζευγών σελίδων (ωφελιμότητα, LRU).

Κεφάλαιο 4

Έξυπνη Διαχείριση τοποθέτησης σελίδων με RNN

4.1 Εισαγωγή	24
4.2 Προσέγγιση με Επαναλαμβανόμενα Νευρωνικά Δίκτυα	24
4.3 Είσοδος στα Επαναλαμβανόμενα Νευρωνικά Δίκτυα	25
4.4 Πρώτη Προσέγγιση – Πρόβλεψη σελίδας που θα αντικατασταθεί από DRAM	26
4.5 Δεύτερη Προσέγγιση – Πρόβλεψη πότε θα αντικατασταθεί η κάθε σελίδα	27
4.6 Τεχνική Υλοποίηση	28

4.1 Εισαγωγή

Αυτό το κεφάλαιο εξετάζει τις προσεγγίσεις μηχανικής νοημοσύνης με RNN που φαίνεται να είναι κατάλληλες κατά το σχεδιασμό ενός δρομολογητή σελίδων για τη διαχείριση σελίδων που έχουν πρόσβαση ή εφαρμογές σε υβριδικά συστήματα μνήμης. Όπως τονίσαμε σε προγενέστερα κεφάλαια, υπάρχει απαίτηση για πιο έξυπνους τρόπους τοποθέτησης σελίδων σε εποχικές περιόδους προγραμματισμού σε διαφορετικές μονάδες μνήμης. Αυτό γίνεται διότι ο δρομολογητής σελίδων που βασίζεται στην πολιτική ιστορικού φαίνεται να είναι ανεπαρκής όσον αφορά την πραγματοποίηση ακριβών προβλέψεων σε ποια μονάδα μνήμης θα τοποθετήσει τις σελίδες.

4.2 Προσέγγιση με Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Μια προσέγγιση μηχανικής μάθησης που είναι κατάλληλη για τη δημιουργία του δρομολογητή σελίδων υβριδικής μνήμης είναι το Recurrent Neural Network (RNN). Το RNN έχει τη δυνατότητα να αναγνωρίζει μοτίβα και να κάνει προβλέψεις σχετικά με τη συμπεριφορά δεδομένων σε μεγάλα χρονικά διαστήματα. Αυτή η μορφή τεχνητής νοημοσύνης έχει χρησιμοποιηθεί για την επίλυση πανομοιότυπων προβλημάτων, όπως η προανάκτηση μνήμης υλικού και η διαχείριση μνήμης σε υβριδικά συστήματα μνήμης [8]. Η προσέγγιση με Επαναλαμβανόμενα Νευρωνικά Δίκτυα (RNN) επιτρέπει μια γραμμική αύξηση της πολυπλοκότητας, καθιστώντας την αποτελεσματική αλλά χρονοβόρα.

Όσον αφορά τον δρομολογητή σελίδων, τα δεδομένα θα μπορούσαν να αναφέρονται στη διαδοχή των σελίδων που προσπελάστηκαν κατά τη διάρκεια της περιόδου εκτέλεσης μιας εφαρμογής. Για να μάθουμε το μοτίβο πρόσβασης σελίδας, θα μπορούσαμε να εφαρμόσουμε ένα RNN στον δρομολογητή σελίδων. Επιπλέον ο δρομολογητής σελίδων μπορεί να εκχωρήσει τις σελίδες που ταξινομεί ο αλγόριθμος πρόβλεψης ως πιο συχνά προσβάσιμες σε μια μονάδα μνήμης χαμηλής καθυστέρησης πρόσβασης τη DRAM. Ενώ οι υπόλοιπες σελίδες μπορούν να αντιστοιχιστούν σε μια μονάδα μνήμης με υψηλότερη καθυστέρηση πρόσβασης την NVM.

Η ακολουθία εισόδου χωρίζεται σε υποακολουθίες μήκους ιστορικού με τη χρήση κυλιόμενου παραθύρου. Κατά τη διάρκεια μιας επανάληψης, κάθε τμήμα δίνεται στο δίκτυο, το οποίο στη συνέχεια προσεγγίζει μια τιμή για κάθε υπό ακολουθία. Χρησιμοποιώντας τη συνάρτηση απώλειας ως βασική παράμετρο, γίνεται μια σύγκριση των πραγματικών και των προβλεπόμενων τιμών έτσι ώστε να παρατηρηθούν οι διαφορές και στη συνέχεια αναπαράγονται για να αλλάξουν τα βάρη. Όταν η μείωση της απώλειας μειώνεται, η εκπαίδευση καταλήγει στο ότι υπάρχει πολύ μικρό κέρδος στην ακρίβεια.

4.3 Είσοδος στα Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Η μελέτη μας καθόρισε ότι τα επαναλαμβανόμενα νευρωνικά δίκτυα είναι τα καταλληλότερα για την πρόκληση που σκοπεύουμε να αντιμετωπίσουμε. Μια ουσιαστική πτυχή της κατασκευής ενός Νευρωνικού Δικτύου είναι η απόφαση για τα χαρακτηριστικά που απεικονίζουν το πρόβλημα και τα οποία μπορούν να χρησιμοποιηθούν ως είσοδοι. Μπορούν να επιδιωχθούν πολυάριθμοι δρόμοι όταν πρόκειται για τη μορφή της ακολουθίας εισόδου του επαναλαμβανόμενου νευρωνικού δικτύου που πρόκειται να δημιουργηθεί. Επιπλέον, θα εξετάσουμε την απεικόνιση της ακολουθίας δεδομένων σχετικά με τη συμπεριφορά πρόσβασης στη μνήμη που θα παρέχεται στο RNN και την αντίληψη της προβλεπόμενης τιμής. Αυτό το βήμα είναι ζωτικής σημασίας όχι μόνο για την ακρίβεια αλλά και για τον χρόνο εκπαίδευσης του ενσωματωμένου μοντέλου.

Όπως αναφέραμε και σε προηγούμενα κεφάλαια τα δεδομένα εισόδου είναι ζωτικής σημασίας και πρέπει να κατανοήσουμε την ποικιλία των δεδομένων που είναι διαθέσιμα για κάθε εφαρμογή. Τα δεδομένα που μπορούν να ληφθούν ενώ μια εφαρμογή βρίσκεται σε λειτουργία ορίζεται ως ένα ίχνος μνήμης. Μια λίστα προσβάσεων σελίδων που λαμβάνονται από την κύρια μνήμη.

Ο μαθησιακός στόχος θα είναι η εκπαίδευσης ενός επαναλαμβανόμενου νευρωνικού δικτύου (RNN) ώστε να μπορεί να προβλέπει μελλοντικές προσβάσεις στη μνήμη σε επίπεδο σελίδας. Όταν εκκινήσει ο δρομολογητής σελίδων, αυτές οι προβλέψεις του RNN θα συγκεντρωθούν για να καθορίσουν την τοποθέτηση σελίδων στις κατάλληλες μονάδες μνήμες (NVM και DRAM).

Θα προσπαθήσουμε να επιταχύνουμε την απόκτηση γνώσεων με περιορισμένους χρόνους εκπαίδευσης. Η προσέγγισή μας που βασίζεται στην μηχανική μάθηση πρέπει να είναι εφαρμόσιμη και να λειτουργεί εντός καθορισμένων χρονικών και υπολογιστικών περιορισμών. Τεχνολογίες όπως GPU, TPU ή εξειδικευμένα RNN μπορούν να επιταχύνουν τη μάθηση, ωστόσο θα πρέπει να βρούμε προσεγγίσεις για να κάνουμε τη γρήγορη μάθηση εφικτή.

4.4 Πρώτη Προσέγγιση – Πρόβλεψη σελίδας που θα αντικατασταθεί από DRAM

Η πιο διαισθητική προσέγγιση ήταν η τροφοδοσία του ίχνους πρόσβασης σελίδων εφαρμογών (workload) σε ένα RNN. Αυτό το RNN επιχειρούσε να προβλέψει τη σελίδα η οποία θα αντικαθίσταται από την DRAM μετά την ακολουθία εισαγωγής που εμπεριείχε τα εξής χαρακτηριστικά: χρόνος N , PageID προσβάσιμης σελίδας το χρόνο N , πίνακας με τα PageIDs των σελίδων που βρίσκονται στην DRAM το χρόνο $N-1$. Προϋπόθεση ήταν ότι η DRAM είχε χωρητικότητα 20 θέσεων σελίδων μόνο. Αυτή η προϋπόθεση εφαρμόστηκε διότι η είσοδος (vector, timeseries) που δημιουργείται για μεγαλύτερη χωρητικότητα DRAM είναι αρκετά μεγάλη με αποτέλεσμα το πρόβλημα να μεγαλώνει εκθετικά.

Για να επιτευχθεί αυτό όσο το δυνατό γρηγορότερα χρησιμοποιήσαμε το PageID για κάθε εικονική διεύθυνση. Αυτό το PageID δημιουργείται βρίσκοντας πρώτα την πραγματική διεύθυνση (Actual Address) αφαιρώντας την ακέραια τιμή της εικονικής διεύθυνση (Virtual Address) στο χρονικό βήμα N από την ακέραια τιμή της εικονικής διεύθυνση $\% 4096$ στο χρονικό βήμα N (ο αριθμός 4096 αναπαριστά το μέγεθος της σελίδας), δηλαδή η εξίσωση:

$$ActualAddress_N = intVirtualAddress_N - (intVirtualAddress_N \% 4096)$$

Και έπειτα δίνουμε στην πραγματική διεύθυνση ένα μοναδικό κωδικό. Αυτό το κάναμε ώστε να μειωθεί ο αριθμός των κατηγοριών που πρέπει να προβλέψει το δίκτυο μας. Επίσης η εξίσωση που αναφέρθηκε προηγουμένως μπορεί να εφαρμοστεί για τη μετάφραση εικονικών

διευθύνσεων σε πραγματικές διευθύνσεις (4KB). Ένα παράδειγμα κατανόησης φαίνεται στον Πίνακα 4.1.

Εικονικές Διευθύνσεις	Πραγματικές Διευθύνσεις	Κωδικός Σελίδας
0x7f132ee02e80	139720367546368	0
0x7f132ee03000	139720367550464	1
0x7f132ee039e8	139720367550464	1

Πίνακας 4.1 Μετατροπή εικονικών διευθύνσεων σε πραγματικές διευθύνσεις και έπειτα απλοποίηση τις πραγματικής διεύθυνσης με μοναδικό κωδικό (PageId)

Αυτή η προσέγγιση είχε ορισμένα σημαντικά μειονεκτήματα που πρέπει να ληφθούν υπόψη. Ο χρόνος εκπαίδευσης ήταν πολύ μεγάλος (αρκετές ημέρες). Αλλά το κύριο πρόβλημα με αυτήν την τεχνική ήταν η δυσκολία επίτευξης ικανοποιητικής ακρίβειας πρόβλεψης. Το εύρος τιμών εξόδου ήταν αρκετά εκτεταμένο γεγονός που οδήγησε σε μείωση ακρίβειας των προβλέψεων του RNN (περίπου 15% επιτυχείς προβλέψεις).

4.5 Δεύτερη Προσέγγιση – Πρόβλεψη πότε θα αντικατασταθεί η κάθε σελίδα

Μια εναλλακτική μέθοδος για την αντιμετώπιση του πιο πάνω προβλήματος είναι να κάνουμε προβλέψεις ανά σελίδα όπως πρότειναν οι ερευνητές στο άρθρο [2,3]. Αντί το RNN μας να προσπαθεί να προβλέψει ποια σελίδα θα αντικατασταθεί από την DRAM όπως αναφέραμε στην πρώτη προσέγγιση, θα εστιάζει στην πρόβλεψη πότε θα προσπελαστεί στη συνέχεια μια σελίδα. Για να επιτευχθεί αυτό, μεμονωμένα RNN (ένα RNN ανά σελίδα) πρέπει να εκπαιδεύονται για κάθε σελίδα εφαρμογής. Το RNN ανά σελίδα τροφοδοτείται με την ακολουθία μετρήσεων πρόσβασης για μια σελίδα σε όλες τις εποχές δρομολόγησης (εποχικές περιόδους προγραμματισμού), από τις οποίες μπορεί στη συνέχεια να προβλέψει τον αριθμό των προσβάσεων που θα λάβει αυτή η σελίδα στην επόμενη περίοδο δρομολόγησης. Αυτή η τροποποίηση στην ακολουθία εισόδου προσφέρει πολλά πλεονεκτήματα.

Η προσέγγιση πρόβλεψης ανά σελίδα είναι μια κατάλληλη λύση στο πρόβλημά μας, το οποίο είναι να ταξινομήσουμε τις σελίδες με βάση τη «θερμότητά» τους χρησιμοποιώντας τον αριθμό πρόσβασής τους εντός χρονικών διαστημάτων δρομολόγησης, προκειμένου να τακτοποιήσουμε τις συγκεκριμένες σελίδες οι οποίες έχουν πιο συχνές προσβάσεις και να οργανώσουμε τη μετεγκατάσταση αυτών των σελίδων στην κατάλληλη μονάδα μνήμης.

Μετά τον μετασχηματισμό του ίχνους δεδομένων, θα χρησιμοποιηθούν Επαναλαμβανόμενα Νευρωνικά Δίκτυα (RNN) προκειμένου να αποκτηθούν πληροφορίες σχετικά με τον αριθμό πρόσβασης κάθε σελίδας σε χρονικά διαστήματα δρομολόγησης.

Τα έξοδα εκπαίδευσης είναι χαμηλά με την πρόβλεψη ανά σελίδα, η οποία απαιτεί ξεχωριστό μοντέλο RNN που πρέπει να εκπαιδεύεται για κάθε σελίδα. Αυτό μοιάζει με μόνο ένα μοντέλο RNN που δημιουργεί προβλέψεις σε όλες τις σελίδες της εφαρμογής, ακόμη και όταν υπάρχουν εκατοντάδες χιλιάδες σελίδες που πρέπει να ληφθούν υπόψη. Συνήθως το μέγεθος του προβλήματος εισόδου παραμένει αμετάβλητο. Είχε διαπιστωθεί προηγουμένως ότι δεν είναι όλες οι σελίδες κρίσιμες για την απόδοση της εφαρμογής. Ο αριθμός των μοντέλων RNN και ο συνολικός χρόνος εκπαίδευσης μπορούν επομένως να ελαχιστοποιηθούν με την εκπαίδευση μεμονωμένων μοντέλων RNN για σελίδες που είναι κρίσιμες για την εφαρμογή.

Μπορεί να ειπωθεί με μεγάλη βεβαιότητα ότι ο τυπικός αριθμός προσβάσεων σελίδας ανά εποχή είναι πολύ χαμηλότερος από το ποσό που θα έπρεπε να εκτιμηθεί προκειμένου να προβλεφθεί η συμπεριφορά σε πολλές σελίδες. Αυτό το μικρότερο εύρος είναι κατάλληλο για εκπαίδευση από RNN, τα οποία γενικά προβλέπουν με υψηλό επίπεδο ακρίβειας. Ως εκ τούτου, είναι πιθανό ότι αυτό θα οδηγήσει σε υψηλή ακρίβεια της πρόβλεψης.

4.6 Τεχνική Υλοποίηση

Πριν από την εμβάθυνση στην περίπλοκη αρχιτεκτονική των Νευρωνικών Δικτύων, όπως τα επίπεδα δικτύου και οι συναρτήσεις απώλειας, είναι σημαντικό να επεκταθούν οι απαραίτητες τεχνικές χειρισμού δεδομένων που θα χρησιμοποιηθούν τόσο για την εκπαίδευση όσο και για τα δεδομένα εισόδου επικύρωσης.

Όπως αναφέρθηκε προηγουμένως, θα εκπαιδεύσουμε και θα επικυρώσουμε ένα επαναλαμβανόμενο νευρωνικό δίκτυο για κάθε επιλεγμένη σελίδα, με βάση τον αντίκτυπό της στην απόδοση εφαρμογής (DRAM hit rate).

Όπως προτείνεται στη βιβλιογραφία [2,3], θα βασιστούμε σε μια ακολουθία μετρήσεων πρόσβασης ανά σελίδα κατά τη διάρκεια διαδοχικών εποχών προγραμματισμού. Η επίτευξη υψηλής ακρίβειας πρόβλεψης με βάση μόνο τον αριθμό πρόσβασης μιας σελίδας αποδεικνύεται αρκετά δύσκολη.

Η έξοδος του επαναλαμβανόμενου νευρωνικού δικτύου ανά σελίδα είναι ο προβλεπόμενος αριθμός προσβάσεων που θα λάβει η σελίδα κατά τη διάρκεια της επόμενης δρομολόγησης.

Κανονικοποιούμε την ακολουθία εισόδου εντός της περιοχής από 0 έως 1, καθώς τα επαναλαμβανόμενα νευρωνικά δίκτυα (ιδιαίτερα τα LSTM) επιδεικνύουν βελτιωμένη απόδοση σε τέτοιες περιπτώσεις όπως γνωρίζουμε. Στη συνέχεια, αποκανονικοποιούμε τα δεδομένα για την τελική πρόβλεψη.

Το δίκτυο περιλαμβάνει δύο στοιβαγμένα επίπεδα LSTM, το καθένα από τα οποία περιέχει 128 νευρώνες, ακολουθούμενα από ένα επίπεδο εξόδου (dense layer). Το μήκος του ιστορικού ορίζεται στο 16, πράγμα που σημαίνει ότι η σειρά δεδομένων εισόδου χωρίζεται σε ακολουθίες των 16 με τον τρόπο του κυλιόμενου παραθύρου. Το σύνολο δεδομένων χωρίζεται, με τα 70% να διατίθενται για εκπαίδευση και το υπόλοιπο 30% για επικύρωση και αξιολόγηση του μοντέλου.

Μια κρίσιμη παράμετρος του Νευρωνικού Δικτύου είναι η επιλογή της συνάρτησης απώλειας η οποία είναι Categorical Crossentropy. Η χρήση δημοφιλών συναρτήσεων απώλειας, όπως η μέση τετραγωνική απώλεια, οδήγησε σε μη ικανοποιητική σύγκλιση ή ακρίβεια λόγω ανισορροπιών δεδομένων.

Το νευρωνικό δίκτυο στοχεύει στην ελαχιστοποίηση αυτής της προσαρμοσμένης τιμής απώλειας μεταξύ προβλεπόμενων και πραγματικών τιμών, χρησιμοποιώντας τον βελτιστοποιητή SGD και ρυθμό εκμάθησης 0,001.

Όσον αφορά την εφαρμογή, τα μοντέλα RNNs πρέπει να είναι συμβατά με την προσομοίωση υβριδικού συστήματος μνήμης που είναι υλοποιημένο στην Python3 [2,3]. Για να το πετύχουμε αυτό, χρησιμοποιούμε την βιβλιοθήκη Keras υψηλού επιπέδου API. Χρησιμοποιούμε την υπάρχουσα υλοποίηση για τους νευρώνες LSTM και τον βελτιστοποιητή SGD για την εκπαίδευση των μοντέλων. Για τυχόν πρόσθετες υπερπαράμετρους που δεν προσδιορίζονται ρητά παραπάνω, εφαρμόζονται οι προεπιλεγμένες τιμές από τη Keras (έκδοση 2.10.0). Η μηχανή εκτέλεσης backend που χρησιμοποιήθηκε είναι η TensorFlow (έκδοση 2.10.0).

Κεφάλαιο 5

Έξυπνη Διαχείριση τοποθέτησης σελίδων με ARIMA(p,d,q)

5.1 Εισαγωγή	30
5.2 Υπόβαθρο χρονολογικών σειρών	30
5.2.1 Ορισμός Χρονολογικής σειράς	31
5.2.2 Στασιμότητα	31
5.3 Υπόβαθρο μοντέλων ARIMA(p, d, q)	32
5.3.1 AR(p) - Αυτοπαλινδρομικό Μοντέλο	33
5.3.2 I(d) - Όρος Διαφοροποίησης	33
5.3.3 MA(q) - Μοντέλο κινούμενου μέσου όρου πεπερασμένης τάξης q	34
5.3.4 Επιλογή παραμέτρων p, d, q	34
5.4 Προσέγγιση – Πρόβλεψη πότε θα αντικατασταθεί η κάθε σελίδα	35
5.5 Τεχνική Υλοποίηση	36

5.1 Εισαγωγή

Τα Αυτοπαλίνδρομα Ολοκληρωμένα Υποδείγματα Κινούμενου Μέσου Τάξης (p,d,q) είναι μια δημοφιλής μέθοδος πρόβλεψης χρονοσειρών [1,10] που συμβολίζεται ως ARIMA(p,d,q). Αυτή η μέθοδος συνδυάζει τα εξής βασικά στελέχη: το μοντέλο Αυτοπαλινδρόμησης (Autoregressive model, συμβολισμός: AR), τον όρο Διαφοροποίησης (Integrated, συμβολισμός: I) και το μοντέλο Κινητού Μέσου Όρου (Moving Average, συμβολισμός: MA).

Τα μοντέλα ARIMA λαμβάνουν υπόψη τη συσχέτιση μεταξύ των παρατηρήσεων στο χρόνο, επιτρέποντας την ανάλυση και την πρόβλεψη μελλοντικών τιμών με βάση τις παρελθούσες τιμές της χρονοσειράς. Η σημασία των μοντέλων ARIMA στην πρόβλεψη χρονοσειρών οφείλεται στην ικανότητά τους να μοντελοποιούν μια ποικιλία χρονοσειρών με διαφορετικά χαρακτηριστικά. Χρησιμοποιώντας τις παραμέτρους p (παράμετρος αυτοπαλινδρόμου υποδείγματος AR), d (ο αριθμός των διαφορών που είναι απαραίτητη για να γίνει μια χρονοσειρά στάσιμη), και q (παράμετρος του υποδείγματος κινητού μέσου όρου MA), το

μοντέλο μπορεί να προσαρμοστεί για να αντιμετωπίσει τάσεις, κυκλικότητα, και άλλα μοτίβα που εμφανίζονται σε χρονοσειρές.

Το μοντέλο ARIMA έχει εφαρμοστεί επιτυχώς σε πολλά πεδία, όπως η οικονομία, η χρηματοοικονομική αγορά [10], η παραγωγή και η διαχείριση εφοδιαστικής αλυσίδας και άλλα, όπου η πρόβλεψη χρονοσειρών είναι ζωτικής σημασίας. Χάρη στην ευελιξία του, το μοντέλο ARIMA μπορεί να προσαρμοστεί και να βελτιωθεί για να αντιμετωπίσει μια ευρεία γκάμα προκλήσεων που σχετίζονται με την πρόβλεψη χρονοσειρών.

Επιπλέον, με τη χρήση τεχνικών μηχανικής μάθησης και βελτιστοποίησης [1], όπως οι αυτόματες μέθοδοι επιλογής παραμέτρων, το μοντέλο ARIMA μπορεί να βελτιωθεί περαιτέρω, παρέχοντας ακόμη πιο ακριβείς και αξιόπιστες προβλέψεις.

Η σημασία του μοντέλου ARIMA στην πρόβλεψη χρονοσειρών είναι αδιαμφισβήτητη και συνεχίζει να αποτελεί ένα βασικό εργαλείο για τους ειδικούς στην ανάλυση δεδομένων. Η κατανόηση και η εφαρμογή του μοντέλου ARIMA μπορεί να οδηγήσει σε βελτιωμένες αποφάσεις, επιτυχή προβλήματα και αυξημένη αποτελεσματικότητα σε πολλούς τομείς της έρευνας και της βιομηχανίας.

5.2 Υπόβαθρο χρονολογικών σειρών

Πριν προχωρήσουμε με την περεταίρω ανάλυση των μοντέλων ARIMA, θα αναλύσουμε μερικές κρίσιμες ορολογίες και χαρακτηριστικά των χρονοσειρών όπου θα τα χρησιμοποιήσουμε για την δημιουργία και την κατανόηση των μοντέλων αυτών.

5.2.1 Ορισμός Χρονολογικής σειράς

Μια χρονολογική σειρά είναι μια σειρά από δεδομένα που ελήφθησαν σε διαδοχικές χρονικές στιγμές. Συνήθως, η χρονοσειρά είναι μια σειρά από μετρήσεις που λαμβάνονται σε συνεχή χρονικά διαστήματα, όπως λεπτά, ώρες, μέρες, μήνες, έτη, κτλ.

Τα δεδομένα της χρονοσειράς χρησιμοποιούνται για την ανάλυση των τάσεων και των μοτίβων που εμφανίζονται στα δεδομένα με την πάροδο του χρόνου. Αυτή η ανάλυση μπορεί να βοηθήσει στην πρόβλεψη ή την πρόγνωση μελλοντικών τιμών ή τάσεων, κάτι που είναι ιδιαίτερα σημαντικό σε πολλές εφαρμογές, όπως η οικονομική ανάλυση, η πρόβλεψη πωλήσεων, οι μετεωρολογικές προβλέψεις, και άλλα.

5.2.2 Στασιμότητα

Μια χρονοσειρά θεωρείται "στάσιμη" όταν οι βασικές στατιστικές της, όπως ο μέσος όρος, η διακύμανση και η αυτοσυσχέτιση, παραμένουν σταθερές με την πάροδο του χρόνου. Αυτό σημαίνει ότι δεν παρουσιάζει τάση ή εποχιακότητα.

Οι όροι "τάση" και "εποχικότητα" αναφέρονται σε δύο κοινά χαρακτηριστικά που μπορεί να παρουσιάζουν οι χρονοσειρές:

- I. Με τον όρο τάση αναφερόμαστε στη μακροχρόνια πορεία μιας χρονοσειράς. Μπορεί να είναι αύξηση, μείωση ή μπορεί να είναι σταθερή. Για παράδειγμα, αν οι τιμές μιας χρονοσειράς αυξάνονται σταθερά με το χρόνο, τότε η χρονοσειρά παρουσιάζει μια θετική τάση. Αντίθετα, αν οι τιμές μειώνονται σταθερά, τότε η τάση είναι αρνητική.
- II. Με τον όρο εποχικότητα αναφερόμαστε σε μοτίβα ή κυκλικές διακυμάνσεις που επαναλαμβάνονται σταθερά μέσα σε συγκεκριμένο χρονικό διάστημα. Για παράδειγμα, μια επιχείρηση που πωλάει παγωτά μπορεί να δει αύξηση των πωλήσεων της κατά τους καλοκαιρινούς μήνες κάθε χρόνο - αυτό θα αποτελούσε μια εποχική διακύμανση.

Αυτές οι παράμετροι μπορούν να επηρεάσουν τη συμπεριφορά μιας χρονοσειράς, και γι' αυτό είναι σημαντικό να τις λάβουμε υπόψη όταν αναλύουμε και προβλέπουμε χρονοσειρές. Οι μέθοδοι πρόβλεψης χρονοσειρών, όπως η μέθοδος ARIMA λαμβάνει υπόψη την τάση και την εποχικότητα όταν κάνει προβλέψεις. Ο αλγόριθμος ARIMA μπορεί να διαχειριστεί τάσεις μέσω του διαφορικού τμήματος (το "I" στο ARIMA, το οποίο σημαίνει "Integrated"). Σε αυτό το τμήμα, ο αλγόριθμος λαμβάνει τη διαφορά μεταξύ διαδοχικών παρατηρήσεων για να αφαιρέσει την τάση. Επίσης, μπορεί να διαχειριστεί την εποχικότητα μέσω εποχιακών μοντέλων ARIMA [9] (πχ. SARIMA και ARIMAX), όπου η εποχικότητα λαμβάνεται υπόψη όταν κάνουμε προβλέψεις.

Τέλος, η εποχικότητα και η τάση μπορεί να είναι και οι δύο αρκετά πολύπλοκες, και ενίοτε ενδέχεται να χρειαστεί πιο προηγμένης μορφής μοντελοποίηση για να τις διαχειριστούμε.

Πιο αναλυτικά, μια χρονοσειρά $\{x_t, x_{t+1}, \dots, x_s\}$ είναι στάσιμη εάν ικανοποιεί τα εξής τρία κριτήρια:

1. Σταθερή μέση τιμή: Ο μέσος όρος της χρονοσειράς δεν πρέπει να αλλάζει με το χρόνο. Αυτό σημαίνει ότι $E(x_t) = \mu_x$ για κάθε t .
2. Σταθερή διακύμανση: Η διακύμανση της χρονοσειράς πρέπει να παραμένει σταθερή με το χρόνο. Αυτό σημαίνει ότι $Var(x_t) = E[x_t - E(x_t)]^2 = \sigma_x^2$, για κάθε t .
3. Σταθερή αυτό συσχέτιση: Η συσχέτιση μεταξύ δύο χρονικών διαστημάτων πρέπει να εξαρτάται μόνο από τη διαφορά των διαστημάτων και όχι από την πραγματική χρονική στιγμή. Δηλαδή, $Cov(x_t + x_{t+k} = Cov(x_{t+m}, x_{t+m+k}) = x_k)$, για κάθε t , όπου k είναι το χρονικό διάστημα.

Εάν μια χρονοσειρά δεν είναι στάσιμη, μπορούν να χρησιμοποιηθούν διάφοροι μέθοδοι για να τη μετατρέψουν σε στάσιμη. Αυτοί οι μέθοδοι μπορούν να περιλαμβάνουν την εφαρμογή διαφορών (λαμβάνοντας τη διαφορά μεταξύ διαδοχικών παρατηρήσεων), την εφαρμογή μετασχηματισμού (όπως ο μετασχηματισμός Box-Cox) ή την εποχιακή προσαρμογή [9] (αφαιρώντας την εποχικότητα).

5.3 Υπόβαθρο μοντέλων ARIMA(p,d,q)

Τώρα σε αυτή την ενότητα είμαστε έτοιμοι να δούμε πιο αναλυτικά τα συστατικά στοιχεία των μοντέλων ARIMA.

5.3.1 AR(p) - Αυτοπαλινδρομικό Μοντέλο

Αναφέρεται στη συσχέτιση μιας τιμής με προηγούμενες τιμές της ίδιας χρονολογικής σειράς. Στο πλαίσιο του μοντέλου ARIMA, ο όρος AR χρησιμοποιείται για να εκφράσει τη σχέση μεταξύ της τρέχουσας τιμής και των προηγούμενων τιμών, βασισμένο στην υπόθεση ότι οι προηγούμενες p τιμές μπορούν να χρησιμοποιηθούν για να προβλέψουν την τρέχουσα τιμή.

Ορίζετε με την εξής εξίσωση: $X_t = \beta_0 + \beta_1 X_{t-1} + \dots + \beta_p X_{t-p} + \varepsilon_t$

– β_0 : αποτελεί την σταθερά μέση τιμή της χρονολογικής σειράς X_t .

– β_i : είναι η επιρροή της X_{t-i} του παρελθόντος στην τρέχουσα X_t .

– ε_t : είναι το σφάλμα το οποίο αποτελείται από ασυσχέτιστες και ισοδύναμες τυχαίες μεταβλητές με μέση τιμή ίση με το μηδέν και σταθερή διακύμανση σ_ε^2 .

5.3.2 I(d) – Όρος Διαφοροποίησης

Ο όρος διακύμανσης/ολοκλήρωσης (I) στο μοντέλο ARIMA αναφέρεται στη διαδικασία διαφοροποίησης της χρονοσειράς, που συνήθως χρησιμοποιείται για να μετατρέψει μια μη στάσιμη χρονοσειρά σε μια στάσιμη. Η διαφοροποίηση υπολογίζει τη διαφορά μεταξύ διαδοχικών τιμών της χρονοσειράς. Η παράμετρος d αντιπροσωπεύει τον αριθμό των φορών που η χρονοσειρά πρέπει να διαφοροποιηθεί για να γίνει σταθερή. Η εφαρμογή της διαφοροποίησης βοηθά στην απομάκρυνση τυχόν τάσεων ή μοτίβων που μπορεί να παρουσιάζονται στη χρονοσειρά καθώς και την προσαρμογή του μοντέλου σε μια πιο κατάλληλη μορφή για την πρόβλεψη.

5.3.3 MA(q) - Μοντέλο κινούμενου μέσου όρου πεπερασμένης τάξης q

Αφορά τη χρήση των παρελθόντων σφαλμάτων πρόβλεψης για να εκτιμήσει την τρέχουσα τιμή. Στα μοντέλα ARIMA, το μοντέλο MA λαμβάνει υπόψη το σφάλμα πρόβλεψης από τα προηγούμενα βήματα χρονικής διακύμανσης, καθώς και το βάρος που αποδίδεται σε κάθε σφάλμα. Η παράμετρος q αντιπροσωπεύει τον αριθμό των παρελθόντων σφαλμάτων που λαμβάνονται υπόψη.

Ορίζετε με την εξής εξίσωση: $X_t = \varepsilon_t + w_1 \varepsilon_{t-1} + w_2 \varepsilon_{t-2} + \dots + w_q \varepsilon_{t-q}$

5.3.4 Επιλογή παραμέτρων p, d, q

Η επιλογή των κατάλληλων τιμών για τους παραμέτρους p , d , και q είναι ζωτικής σημασίας για την απόδοση και την ακρίβεια του μοντέλου ARIMA. Αυτή η διαδικασία συνήθως περιλαμβάνει τη χρήση διαφόρων στατιστικών κριτηρίων, όπως το Akaike Κριτήριο Πληροφορίας (Akaike Information Criterion, AIC) ή το Bayesian Κριτήριο Πληροφορίας (Bayesian Information Criterion, BIC) [9], για να συγκρίνουν διαφορετικά μοντέλα και να επιλέξουν το βέλτιστο μοντέλο.

Είναι επίσης σημαντικό να εξετάσουμε τη στατιστική σημασία των παραμέτρων του μοντέλου, καθώς και την αυτό συσχέτιση των υπολειμμάτων, ώστε να εξασφαλίσουμε ότι το τελικό μοντέλο παρέχει μια καλή προσαρμογή στα δεδομένα και είναι σε θέση να προβλέψει τη χρονοσειρά με ακρίβεια.

Συνοψίζοντας, το μοντέλο ARIMA είναι μια δημοφιλής και ευέλικτη μέθοδος πρόβλεψης χρονοσειρών που βασίζεται στην ιδέα της αυτοπαλίνδρομης, της ολοκλήρωσης και του

κινούμενου μέσου. Η ανάλυση των παραμέτρων p , d , και q καθώς και η χρήση στατιστικών κριτηρίων και ελέγχων, μπορεί να βοηθήσει στην επιλογή του κατάλληλου μοντέλου που ταιριάζει καλύτερα στα δεδομένα και παρέχει ακριβείς προβλέψεις.

Το μοντέλο ARIMA είναι ένα ισχυρό και ευέλικτο εργαλείο για την πρόβλεψη χρονοσειρών, αλλά όπως και με οποιοδήποτε άλλο μοντέλο, η αποτελεσματικότητά του εξαρτάται από την επιλογή των κατάλληλων παραμέτρων και την προσαρμογή του στα συγκεκριμένα δεδομένα. Ως εκ τούτου, η διαδικασία της παραμετροποίησης (εκπαίδευσης) ενός μοντέλου ARIMA μπορεί να είναι περίπλοκη και χρονοβόρα, ενώ η επιλογή του κατάλληλου μοντέλου εξαρτάται σε μεγάλο βαθμό από την εμπειρία και την κρίση του κάθε αναλυτή/ερευνητή.

Επιπρόσθετα, το ARIMA δεν είναι ιδανικό για όλες τις περιπτώσεις χρονοσειρών. Για παράδειγμα, μπορεί να μην είναι κατάλληλο για την πρόβλεψη χρονοσειρών με πολύπλοκες μη γραμμικές συσχετίσεις ή αυξανόμενες τάσεις. Σε τέτοιες περιπτώσεις, άλλα μοντέλα όπως τα νευρωνικά δίκτυα, τα μοντέλα προσομοίωσης ή άλλες μη γραμμικές μεθόδους μπορεί να προσφέρουν καλύτερα αποτελέσματα.

Συνοψίζοντας, το μοντέλο ARIMA είναι μια δημοφιλής και ευρέως χρησιμοποιούμενη μέθοδος για την πρόβλεψη χρονοσειρών [9,10,1]. Ενώ προσφέρει αξιόλογα αποτελέσματα σε πολλές περιπτώσεις, οι αναλυτές πρέπει να είναι προσεκτικοί κατά την επιλογή των παραμέτρων και να εξετάσουν εναλλακτικά μοντέλα, όταν το ARIMA δεν προσφέρει ικανοποιητικά αποτελέσματα.

Στο πλαίσιο αυτό, η συνεχής έρευνα και η ανάπτυξη νέων τεχνικών και μεθοδολογιών είναι κρίσιμη για τη βελτίωση της προσέγγισης των προβλέψεων χρονοσειρών. Αυτό περιλαμβάνει την κατανόηση των περιορισμών και των πλεονεκτημάτων των διαφόρων μοντέλων, την εξερεύνηση νέων μεθόδων για την επιλογή παραμέτρων και την εξέταση συνδυαστικών μοντέλων που συμπεριλαμβάνουν πολλαπλές προσεγγίσεις για την πρόβλεψη χρονοσειρών.

5.4 Προσέγγιση – Πρόβλεψη πότε θα αντικατασταθεί η κάθε σελίδα

Όπως αναφέραμε και στο Κεφάλαιο 4 στην υπό ενότητα 4.5 θα κάνουμε προβλέψεις ανά σελίδα. Δηλαδή θα προσπαθήσουμε να προβλέψουμε πότε θα προσπελαστεί στη συνέχεια μια σελίδα χρησιμοποιώντας ένα αλγόριθμο ARIMA. Για να επιτευχθεί αυτό, μεμονωμένα ARIMA (ένα ARIMA ανά σελίδα) πρέπει να εκπαιδεύονται για κάθε ωφέλιμη σελίδα εφαρμογής που μας ενδιαφέρει (ακριβώς τις ίδιες σελίδες που θα χρησιμοποιήσουν τα

μοντέλα RNNs). Το ARIMA ανά σελίδα τροφοδοτείται με την χρονολογική σειρά μετρήσεων πρόσβασης για μια σελίδα σε όλες τις εποχές δρομολόγησης (εποχικές περιόδους προγραμματισμού), από τις οποίες στη συνέχεια θα προβλέψει τον αριθμό των προσβάσεων που θα λάβει αυτή η σελίδα στην επόμενη περίοδο δρομολόγησης.

5.5 Τεχνική Υλοποίηση

Εφόσον δημιουργηθεί η χρονοσειρά μας όπως αναφερθήκαμε στην πιο πάνω ενότητα αλλά και σε προγενέστερα κεφάλαια, θα διαιρέσουμε τα δεδομένα μας σε δύο μέρη, το σύνολο εκπαίδευσης που αποτελεί το 70% της χρονοσειράς (όπου σε αυτό θα πρέπει να βρούμε τις κατάλληλες παραμέτρους p , d , q) και το σύνολο δοκιμής που αποτελεί το εναπομείναντα 30% της χρονοσειράς.

Όπως είδαμε σε αυτό το κεφάλαιο η επιλογή των κατάλληλων παραμέτρων p, d, q είναι ζωτικής σημασίας για την εφαρμογή και την πρακτικότητα του μοντέλου ARIMA. Η επιλογή αυτών των παραμέτρων πραγματοποιείται χρησιμοποιώντας την συνάρτηση `auto_arima()` από τη βιβλιοθήκη `pmдарima` στην επιλεγμένη χρονοσειρά εκπαίδευσης. Η συνάρτηση `auto_arima()` ελέγχει αν η χρονολογική σειρά είναι στάσιμη και αν δεν είναι την προσαρμόζει ώστε για να γίνει. Αυτό γίνεται μέσω του κριτηρίου ADF που ελέγχει αν υπάρχει τάση στην χρονοσειρά. Σημαντική υπόθεση είναι ότι οι χρονοσειρές μας δεν εμφανίζουν εποχικότητα οπότε δεν γίνεται έλεγχος για αυτό. Έπειτα αυτόματα η συνάρτηση επιλέγει τις καλύτερες παραμέτρους με βάση τις τιμές που έχουν προσδιοριστεί για τα όρια των παραμέτρων (`start_p`, `start_q`, `max_p`, `max_q`) και τη μέθοδο επιλογής (`stepwise`).

Μετά την επιλογή των βέλτιστων παραμέτρων, το μοντέλο ARIMA δημιουργείται και προσαρμόζεται με βάση αυτές τις παραμέτρους και έπειτα πραγματοποιούνται προβλέψεις σε όλο το σύνολο των δεδομένων.

Αυτές οι προβλέψεις μετατρέπονται σε κατηγορίες σύμφωνα με το όριο ± 10 όλων των αποδεκτών προβλέψεων (κατηγοριοποίηση). Αυτό γίνεται διότι οι προβλέψεις του μοντέλου ARIMA είναι δεκαδικές και θα πρέπει να τις προσεγγίσουμε σαν ακέραιες τιμές βάση του προβλήματος που προσπαθούμε να επιλύσουμε, καθώς και γιατί μας νοιάζει η πιο ακριβής πρόβλεψη που μπορούμε να έχουμε. Επεξηγηματικά τα βήματα της πιο πάνω διαδικασίας παρουσιάζονται με το πιο κάτω παράδειγμα:

Έστω η σελίδα με κωδικό 109 που έχει την εξής χρονολογική σειρά:

Αριθμός περιόδου δρομολόγησης	1	2	3	4	5	6
Αριθμός προσβάσεων σελίδας με PageID 109	50	0	60	20	0	60

Τότε θα έχει τις εξής κατηγορίες σαν αποδεκτές προβλέψεις:

0	20	50	60
---	----	----	----

Έστω ότι το ARIMA προσαρμόζεται στις πρώτες 3 περιόδους δρομολόγησης και έπειτα κάνει τις εξής προβλέψεις για τις περίοδο δρομολόγησης 4, 5, 6 αντίστοιχα:

4	5	6
55,023	0,5454	62,565

Τότε οι πραγματικές προβλέψεις πρέπει να είναι 0 ή 20 ή 50 ή 60, έχοντας θέση το όριο ± 10 δημιουργούνται οι εξής κατηγορίες-όρια :

Όρια	$(-10) - 10$	10-30	40-60	50-70
Κατηγορίες	0	20	50	60

Τέλος με σειρά προτεραιότητας από τα δεξιά προς τα αριστερά η πρόβλεψη 55,023 γίνεται 60, η πρόβλεψη 0,5454 γίνεται 0 και η πρόβλεψη 62,565 γίνεται 60.

Επίσης στο τέλος της εφαρμογής του ARIMA υπολογίζονται οι εξής μετρικές απόδοσης: σφάλμα πρόβλεψης και η ακρίβεια πρόβλεψης τιμών.

Κεφάλαιο 6

Αξιολόγηση Αποτελεσμάτων

6.1 Εισαγωγή	38
6.2 Ίχνη εφαρμογών σαν Δεδομένα Εισόδου	39
6.3 Προσομοιωτής Συστήματος Υβριδικής Μνήμης	39
6.4 Ακρίβεια πρόβλεψης μοντέλων RNNs	40
6.5 Ακρίβεια πρόβλεψης μοντέλων ARIMA	41
6.6 Απόδοση Εφαρμογών με χρήση RNNs	42
6.7 Απόδοση Εφαρμογών με χρήση ARIMA	44

6.1 Εισαγωγή

Στο τρέχον κεφάλαιο, διεξάγουμε μια εξέταση της απόδοσης του Δρομολογητή Σελίδων Εφαρμογών βαθμολογώντας το DRAM hit rate για τους 9 φόρτους εργασίας που έχουμε στην διάθεση μας. Όπως αναφέρθηκε στα άρθρα [2,3] καθώς και σε προγενέστερα κεφάλαια αυτής της διπλωματικής εργασίας, ξέρουμε ότι υπάρχει σημαντική βελτίωση της απόδοσης μιας εφαρμογής (DRAM hit rate) ακόμη και όταν διαχειριζόμαστε έξυπνα ένα μικρό ποσοστό των σελίδων ανά εφαρμογή. Αυτό ήταν εφαρμόσιμο δεδομένου ότι είχαμε προφητική επίγνωση (πολιτική oracle) των αριθμών προσβάσεων σελίδων ανά κάθε εποχή δρομολόγησης.

Στόχος μας είναι να προσεγγίσουμε αυτήν την προφητική κατανόηση των μετρήσεων πρόσβασης, χρησιμοποιώντας μηχανική μάθηση και στατιστικά μοντέλα. Έτσι, αυτό το κεφάλαιο επιδιώκει όχι μόνο να μετρήσει τα DRAM hit rates που προσφέρονται από τον Δρομολογητή Σελίδων Εφαρμογών, αλλά και την ακρίβεια πρόβλεψης των Επαναλαμβανόμενων νευρωνικών δικτύων και των στατιστικών μοντέλων ARIMA που αναπτύχθηκαν για ένα μικρό ποσοστών σελίδων για κάθε εφαρμογή.

6.2 Ίχνη εφαρμογών σαν Δεδομένα Εισόδου

Για την αξιολόγηση του Δρομολογητή Σελίδων, χρησιμοποιήσαμε ένα σύνολο τάξεως 9 εφαρμογών. Οι φόρτοι εργασίας που χρησιμοποιήθηκαν ήταν με το PARSEC και το Rodinia 3.1 (benchmark suites) όπως περιγράφονται στα άρθρα [2,3]. Τα δεδομένα (workloads) έχουν παρθεί από την ιστοσελίδα [18].

Αυτά τα ίχνη εφαρμογών αποτελούνται από τις διευθύνσεις εικονικής μνήμης 4KB στις οποίες είχε πρόσβαση η κάθε εφαρμογή. Στη συνέχεια, κατηγοριοποιούμε τις προσβάσεις στη μνήμη σε εποχικές περιόδους προγραμματισμού, ακολουθώντας μια μεθοδολογία παρόμοια με αυτή που χρησιμοποιήσαμε στον δρομολογητή μας στο κεφάλαιο 3.

6.3 Προσομοιωτής Συστήματος Υβριδικής Μνήμης

Στόχος μας είναι να αξιολογήσουμε την απόδοση του Δρομολογητή Σελίδων που έχουμε αναπτύξει. Ιδανικά, ολόκληρη η διαδικασία αξιολόγησης θα διεξαγόταν σε ένα πραγματικό Υβριδικό Σύστημα Μνήμης. Ωστόσο, αυτό είναι επί του παρόντος είναι σχεδόν αδύνατο. Η απουσία ενός API σε επίπεδο συστήματος για τη μετεγκατάσταση σελίδων μνήμης σε διαφορετικά στοιχεία μνήμης και η έλλειψη μιας καθολικά αποδεκτής και καθιερωμένης μεθόδου για την καταγραφή των προσβάσεων κύριας μνήμης μας οδήγησαν να χρησιμοποιήσουμε μια προσαρμοσμένη, Προσομοίωση Υβριδικού Συστήματος Μνήμης όπως προτάθηκε στα άρθρα [2,3] αλλά με πιο απλοϊκή υλοποίηση.

Προσομοιώνουμε ένα υβριδικό σύστημα μνήμης που αποτελείται από δύο διακριτά στοιχεία μνήμης: ένα γρήγορο (π.χ. DRAM) και ένα με υψηλότερη καθυστέρηση πρόσβασης (π.χ. NVM). Το χαρακτηριστικό μόνιμης μνήμης (PM) της NVM δεν λαμβάνεται υπόψη στο περιβάλλον προσομοίωσης μας, καθώς υποθέτουμε ότι το NVM έχει ρυθμιστεί σε λειτουργία μνήμης και χρησιμεύει ως επέκταση της DRAM.

Η αναλογία μεταξύ NVM και DRAM μπορεί να διαμορφωθεί στον προσομοιωτή μας. Ρυθμίσαμε τον προσομοιωτή μας με αναλογία DRAM προς NVM 1:8, αυτό σημαίνει ότι η DRAM θα έχει περιορισμένη χωρητικότητα και δεν μπορεί να φιλοξενήσει περισσότερο από το 1/8 των σελίδων της εφαρμογής, ενώ η πιο αργή NVM θα εξυπηρετεί το άλλα 7/8.

Ο προσομοιωτής μας έχει σχεδιαστεί σύμφωνα με τα άρθρα [2,3] για να φιλοξενεί τον δρομολογητή σελίδων μας, που χρησιμοποιείται κυρίως για τη λήψη πληροφοριών της DRAM hit rate. Μετά την προετοιμασία του Προσομοιωτή Συστήματος Υβριδικής Μνήμης [18] και του Δρομολογητή Σελίδας μας για ένα δεδομένο ίχνος (workload) πρόσβασης στη μνήμη, λαμβάνουμε δεδομένα για το πόσες και ποιες προσβάσεις σελίδας εξυπηρετήθηκαν από την DRAM και ποιες από τη πιο αργή NVM. Επίσης είναι σημαντικό να αναφέρουμε ότι η πολιτική αδειάσματος σελίδας DRAM που χρησιμοποιήθηκε είναι αυτή που αναφέραμε και αναλύσαμε στο κεφάλαιο 3 στην ενότητα 3.3.

6.4 Ακρίβεια πρόβλεψης μοντέλων RNNs

Όπως αναφέρθηκε με έμφαση νωρίτερα στην εισαγωγή του κεφαλαίου αυτού, ένα κρίσιμο στοιχείο που πρέπει να αξιολογήσουμε είναι το ποσοστό επιτυχίας προβλέψεων των Επαναλαμβανόμενων Νευρωνικών Δικτύων (RNNs). Αυτά τα μοντέλα μας επιτρέπουν να κάνουμε προβλέψεις σχετικά με τον αριθμό πρόσβασης των 100 πιο κερδοφόρων σελίδων για την κάθε επόμενη εποχή δρομολόγησης σελίδων.

Για κάθε φόρτο εργασίας/εφαρμογή, χρησιμοποιούμε ένα RNN για κάθε μια από τις 100 πιο κερδοφόρες σελίδες ανά εφαρμογή, όπως προσδιορίζονται από τον Επιλογέα Σελίδας στο κεφάλαιο 3 ενότητα 3.2.1.

Για να αξιολογήσουμε την προγνωστική ακρίβεια των μοντέλων μας, για κάθε μοντέλο RNN μετά το πέρας της εκπαίδευσης του, μετρούσαμε πόσες σωστές προβλέψεις έκανε στα δεδομένα επαλήθευσης συγκρίνοντας την πρόβλεψη με την αναμενόμενη σωστή πρόβλεψη (σύμφωνα με τον Oracle Page Scheduler = πολιτική oracle). Έπειτα υπολογίζαμε τον μέσο όρο επιτυχίας του μοντέλου σε ποσοστό για τα δεδομένα επαλήθευσης. Στο τέλος συσσωρεύαμε όλα τα ποσοστά επιτυχίας για όλα τα RNNs που αναπτύχθηκαν για τις 100 πιο κερδοφόρες/ωφέλιμες σελίδες ανά εφαρμογή και υπολογίζαμε το συνολικό ποσοστό επιτυχίας για όλες τις κερδοφόρες σελίδες της κάθε εφαρμογής. Δηλαδή συνοψίζοντας τα πιο πάνω, μπορούσαμε να πούμε ότι τα 100 RNNs μας για την εφαρμογή χ είχαν την δυνατότητα να προβλέψουν με ένα ποσοστό επιτυχίας τον αριθμό προσβάσεων των 100 ωφέλιμων σελίδων σε όλες τις εποχές δρομολόγησης. Ακολουθεί ο πίνακας 6.1 με τα αποτελέσματα.

ΕΦΑΡΜΟΓΗ	ΠΕΡΙΟΔΟΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΑΝΑ ΟΛΟΚΛΗΡΩΣΗ # ΠΡΟΣΒΑΣΕΩΝ ΣΕΛΙΔΩΝ	# ΠΕΡΙΟΔΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ	ΠΟΣΟΣΤΟ ΑΚΡΙΒΕΙΑΣ ΠΡΟΒΛΕΨΕΩΝ ΤΩΝ RNN
backprop	9000	68	41%
kmeans	26000	20	57%
hotspot	24900	26	89%
quicksilver	31100	36	62%
cpd	35500	26	24%
lud	23200	31	12%
bfs	12400	54	38%
bptree	27600	21	33%
pennant	11700	45	49%

Πίνακας 6.1 Συνολικά αποτελέσματα των 100 RNNs (1 RNN για κάθε ωφέλιμη σελίδα) ανά εφαρμογή.

Η περίοδος δρομολόγησης ορίζεται ως ο αριθμός των προσβάσεων σε σελίδες από την εκάστοτε εφαρμογή. Οι περίοδοι δρομολόγησης που επιλέχθηκαν για την κάθε εφαρμογή δεν είναι τυχαίες αλλά επιλέχθηκαν σύμφωνα με το άρθρο [4] που μας λέει ότι με αυτό τον αριθμό θα χρειαστεί πιο λίγη ώρα εκπαίδευσης για τα RNN (περίπου 30-60 λεπτά).

Παράδειγμα κατανόησης από τον πίνακα: Για την εφαρμογή backprop η δρομολόγηση γίνεται κάθε 9000 προσβάσεις σελίδων από την εφαρμογή, έτσι καταλήγουμε ότι για το φόρτο εργασίας της εφαρμογής θα υπάρχουν 68 περίοδοι δρομολόγησης. Αναπτύσσεται και εκπαιδεύεται ένα RNN για κάθε μια από τις 100 πιο κερδοφόρες σελίδες και καταλήγουμε στο ποσοστό ακρίβειας προβλέψεων τάξεως 41%. Δηλαδή τα 100 RNN's έχουν επιτύχει συνολικά να προβλέψουν σωστά το 41% από όλες τις απαιτούμενες προβλέψεις στα δεδομένα επαλήθευσης των 100 πιο κερδοφόρων σελίδων της εφαρμογής.

Σημαντικό είναι να αναφέρουμε ότι η εφαρμογή hotspot δεν έχει 100 κερδοφόρες σελίδες αλλά μόνο 89. Αυτό δεν μας επηρεάζει κάπου απλώς το αναφέρουμε για σκοπούς ανάλυσης και επαλήθευσης. Περισσότερα και πιο αναλυτικά στοιχεία μπορούμε να τα δούμε στο Παράρτημα Α.

6.5 Ακρίβεια πρόβλεψης μοντέλων ARIMA's

Ακολουθούμε τα ίδια δεδομένα/πληροφορίες που αναφέραμε στην ενότητα 6.4 με την μόνη διαφορά ότι αντί να έχουμε RNN's έχουμε μοντέλα ARIMA's όπως τα μελετήσαμε στο κεφάλαιο 5. Καταλήγουμε στα πιο κάτω αποτελέσματα που φέρονται στον Πίνακα 6.2.

ΕΦΑΡΜΟΓΗ	ΠΕΡΙΟΔΟΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΑΝΑ ΟΛΟΚΛΗΡΩΣΗ # ΠΡΟΣΒΑΣΕΩΝ ΣΕΛΙΔΩΝ	# ΠΕΡΙΟΔΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ	ΠΟΣΟΣΤΟ ΑΚΡΙΒΕΙΑΣ ΠΡΟΒΛΕΨΕΩΝ ΤΩΝ ARIMA
backprop	9000	68	30%
kmeans	24000	33	34%
hotspot	20000	22	19%
quicksilver	31100	36	57%
cpd	35500	26	16%
lud	23200	31	12%
bfs	12400	54	29%
bptree	27600	21	27%
pennant	11700	45	27%

Πίνακας 6.2 Συνολικά αποτελέσματα των 100 ARIMA (1 ARIMA για κάθε ωφέλιμη σελίδα) ανά εφαρμογή.

Η περίοδος δρομολόγησης για κάθε εφαρμογή είναι η ίδια με αυτή που χρησιμοποιήθηκε για τα RNNs για σκοπούς απλότητας και σύγκρισης. Η μόνη διαφορά είναι στις εφαρμογές kmeans και hotspot για το λόγο του ότι οι χρονοσειρές που δημιουργούνται ήταν πολύ μικρές σε μήκος με αποτέλεσμα να παρουσιάζουν εποχικότητα και συνεπώς δεν μπορούσαν να μετατραπούν αυτόματα σε στάσιμες ώστε να αναπτυχθούν τα μοντέλα ARIMA.

Συμπερασματικά έχουμε καλύτερες προβλέψεις με τα μοντέλα RNNs παρά με τα μοντέλα ARIMA.

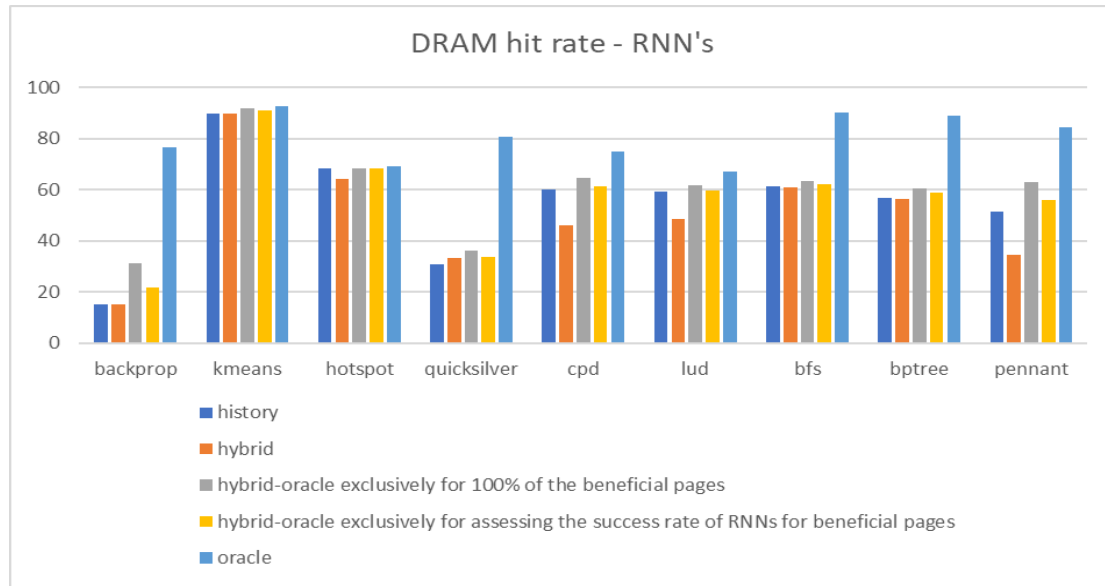
6.6 Απόδοση Εφαρμογών με χρήση RNNs

Για να αξιολογήσουμε την απόδοση εφαρμογών, χρησιμοποιούμε μια ενδεικτική μέτρηση, δηλαδή το ποσοστό επιτυχίας DRAM (δηλαδή το ποσοστό των αιτημάτων σελίδων που εκπληρώνονται από τη DRAM). Ο ρυθμός επιτυχίας DRAM (DRAM hit rate) της υλοποίησής μας μπορεί να προκύψει από την Προσομοίωση Υβριδικού Συστήματος Μνήμης που περιγράφεται στην ενότητα 6.3. Όπως είναι κατανοητό, ένα υψηλότερο ποσοστό επιτυχίας DRAM σημαίνει μια πιο ουσιαστική ώθηση στην απόδοση της εφαρμογής, καθώς περισσότερα αιτήματα σελίδων εφαρμογών εξυπηρετούνται από το σημαντικά ταχύτερο στοιχείο μνήμης.

Πριν αξιολογήσουμε την απόδοση των εφαρμογών (DRAM hit rates) θα αναλύσουμε τι εννοούμε με της εξής πολιτικές:

- Ένας Δρομολογητής Σελίδων βασισμένος στην πολιτική Ιστορικού (history) προβλέπει ότι μια σελίδα θα λάβει τα ίδια μετρήματα πρόσβασης σελίδων στην επόμενη εποχή δρομολόγησης όπως και στην τρέχουσα εποχή δρομολόγησης.
- Εκτός από την πολιτική ιστορίας, έχουμε την πολιτική oracle που όπως ξέρουμε είναι παρόμοια με το να έχουμε ένα παντογνώστη που γνωρίζει από πριν όλες τις προσβάσεις σελίδων για κάθε εποχή δρομολόγησης. Δηλαδή αυτό θα είναι το ιδανικό και βέλτιστο αποτέλεσμα που μπορούμε να έχουμε.
- Η υβριδική πολιτική είναι όταν έχουμε τα RNNs να προβλέπουν τα μετρήματα πρόσβασης των 100 πιο κερδοφόρων σελίδων εφαρμογής για όλες τις εποχές δρομολόγησης σελίδων και για τις υπόλοιπες σελίδες οι προβλέψεις διαχειρίζονται από την πολιτική ιστορικού (αυτή η πολιτική αναπτύσσετε για στατιστικούς σκοπούς και για περαιτέρω μελέτη).
- Μια άλλη πολιτική (hybrid-oracle exclusively for 100% of the beneficial pages) είναι η υβριδική προσέγγιση σε συνδυασμό με μια πολιτική oracle που θα διαχειρίζεται το 100% όλων των πιο κερδοφόρων σελίδων της εφαρμογής και τις υπόλοιπες σελίδες θα της διαχειρίζεται με την πολιτική ιστορικού. Αυτή η πολιτική μας δείχνει το μέγιστο δυνατό DRAM hit rate που μπορούσε να επιτευχθεί αν τα RNNs μας είχαν 100% επιτυχείς προβλέψεις (δηλαδή η 100 πιο ωφέλιμες σελίδες ανά εφαρμογή τοποθετούταν στην σωστή μονάδα μνήμης).
- Η τελευταία πολιτική (hybrid-oracle exclusively for assessing the success rate of RNNs for beneficial pages) που μας ενδιαφέρει περισσότερο είναι η υβριδική προσέγγιση σε συνδυασμό με μια πολιτική oracle που θα διαχειρίζεται το ποσοστό επιτυχίας πρόβλεψης των RNNs των κερδοφόρων σελίδων της εφαρμογής και οι υπόλοιπες σελίδες θα διαχειρίζονται από την πολιτική ιστορικού (δηλαδή αν τα RNNs για μια εφαρμογή είχαν συνολικές επιτυχείς προβλέψεις με ποσοστό 80% για τις 100 πιο ωφέλιμες σελίδες αυτό θα θεωρούμε ότι είναι το ίδιο με το να χειρίζεται τις 80 αυτές σελίδες ο oracle page scheduler).

Εφόσον εξηγήσαμε πως μπορούμε να διαβάσουμε τα αποτελέσματα σύμφωνα με την κάθε πολιτική που ακολουθήθηκε μπορούμε να δούμε τα αποτελέσματα στην πιο κάτω γραφική παράσταση 6.1 (ακριβής μετρήσεις μπορούμε να τις δούμε αναλυτικά στο Παράρτημα Α).

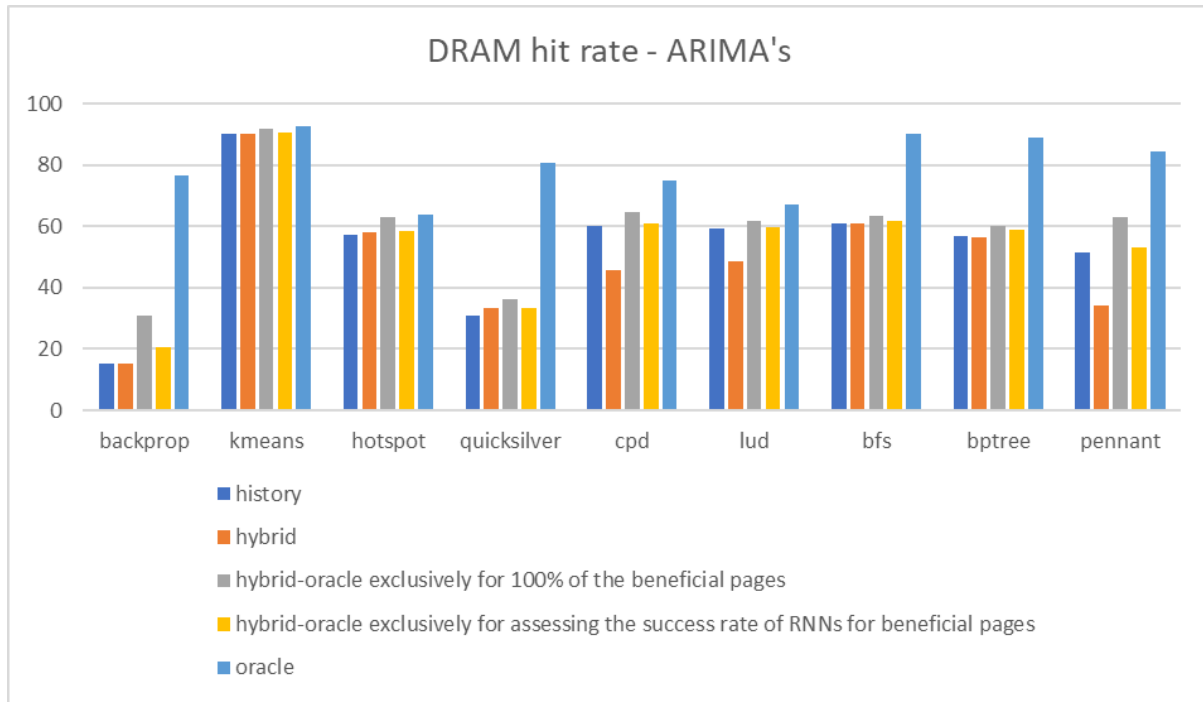


Γραφική Παράσταση 6.1 Παρουσιάζει τα DRAM hit rates για την κάθε εφαρμογή και την κάθε πολιτική έχοντας χρησιμοποιήσει RNNs.

Συνοπτικά, είναι προφανές ότι η προγνωστική ακρίβεια των εκπαιδευμένων RNN είναι τέτοια που μπορεί να προσφέρει απόδοση εφαρμογής παρόμοια με ό,τι θα μπορούσε να επιτευχθεί με προφητική γνώση των αριθμών πρόσβασης στη σελίδα.

6.7 Απόδοση Εφαρμογών με χρήση ARIMA

Παρομοίως ισχύουν όλες οι πληροφορίες/δεδομένα που διατυπώσαμε στην ενότητα 6.6 με την μόνη διαφορά ότι όπου μιλάμε για RNNs ισχύει το ίδιο για τα μοντέλα ARIMA. Μπορούμε να δούμε τα αποτελέσματα στην πιο κάτω γραφική παράσταση 6.2 (ακριβής μετρήσεις μπορούμε να τις δούμε αναλυτικά στο Παράρτημα Β).



Γραφική Παράσταση 6.2 Παρουσιάζει τα DRAM hit rates για την κάθε εφαρμογή και την κάθε πολιτική έχοντας χρησιμοποιήσει μοντέλα ARIMA.

Όπως αναμενόταν η καλύτερη απόδοση εφαρμογών είναι με την χρήση μοντέλων RNNs παρά με την χρήση μοντέλων ARIMA.

Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα έρευνας	46
7.2 Προτάσεις για μελλοντική έρευνα	47

7.1 Συμπεράσματα έρευνας

Το αυξανόμενο μέγεθος των δεδομένων εφαρμογών και η αυξανόμενη πολυπλοκότητα των μοτίβων πρόσβασης σελίδων θέτουν σημαντικές προκλήσεις για τις συμβατικές τεχνολογίες υλικού μνήμης. Αυτές οι τεχνολογίες συχνά αγωνίζονται να προσφέρουν την απαιτούμενη χωρητικότητα και βελτιώσεις ταχύτητας για την υποστήριξη των σύγχρονων εφαρμογών. Κατά συνέπεια, η ενσωμάτωση καινοτόμων τεχνολογιών υλικού στα υποσυστήματα μνήμης καθίσταται κρίσιμη για την αύξηση της απόδοσης της κάθε εφαρμογής και τη βελτίωση της αποδοτικότητας κόστους στο σύστημα.

Ωστόσο, οι υπάρχουσες στρατηγικές διαχείρισης πόρων σε επίπεδο συστήματος δεν είναι καλά εξοπλισμένες για την πλοήγηση στην αναδυόμενη ετερογένεια στις συμπεριφορές πρόσβασης σελίδων στο υλικό μνήμης και στις εφαρμογές. Υπάρχει ένα κενό μεταξύ της απόδοσης που επιτυγχάνεται μέσω των υφιστάμενων λύσεων στη διαχείριση του Υβριδικού συστήματος μνήμης (Hybrid Memory System) και των δυνατοτήτων βέλτιστης απόδοσης. Σε αυτήν την έρευνα, εμβαθύνουμε στη σκοπιμότητα και την αποτελεσματικότητα της χρήσης μηχανικής μάθησης και των στατιστικών μοντέλων στη Διαχείριση Μνήμης ενός προσομοιωτή υβριδικού συστήματος.

Λαμβάνοντας υπόψη ότι οι εφαρμογές που εκτελούνται σε υβριδικά συστήματα μνήμης είναι πιθανό να έχουν σημαντικά αποτυπώματα μνήμης, ο δρομολογητής σελίδων μας εντοπίζει ένα επιλεγμένο υποσύνολο σελίδων. Η διαχείριση αυτών των σελίδων μέσω μηχανικής μάθησης και στατιστικών μοντέλων ενισχύει την απόδοση της εφαρμογής. Στη συνέχεια, καθεμία από αυτές τις επιλεγμένες σελίδες αντιμετωπίζεται από ένα επαναλαμβανόμενο νευρωνικό δίκτυο και από ένα στατιστικό μοντέλο ARIMA, σχεδιασμένο να μαθαίνει μοτίβα πρόσβασης σε επίπεδο σελίδας. Για τις υπόλοιπες σελίδες, καταφεύγουμε σε μια πολιτική βασισμένη στο ιστορικό.

Συνοψίζοντας καταλήγουμε ότι την καλύτερη απόδοση προβλέψεων την έχουμε με την χρήση RNNs σε σύγκριση με τα μοντέλα ARIMA. Τα αποτελέσματα μας θα ήταν πολύ καλύτερα και θα έδειχναν μια τεράστια διαφορά απόδοσης (DRAM hit rates) αν οι περίοδοι δρομολόγησής ήταν ≥ 1000 (αυτό όμως θα συνεπαγόταν με πολλές μέρες εκπαίδευσης για την κάθε εφαρμογή και θα χρειαζόταν μεγάλη υπολογιστική δύναμη) [2,3].

7.2 Προτάσεις για μελλοντική έρευνα

Οι γνώσεις που προέκυψαν από την αξιολόγηση αυτής της εργασίας στο κεφάλαιο 6, σε συνδυασμό με τις υποκείμενες υποθέσεις και τις απλοποιήσεις που χρησιμοποιήθηκαν κατά την υλοποίησή της, ανοίγουν μια πληθώρα οδών για μελλοντική έρευνα. Υπάρχουν πολυάριθμες ιδέες και κατευθύνσεις που θα μπορούσαν να επιδιωχθούν για να ενισχύσουν και να εμβαθύνουν το έργο που παρουσιάζεται σε αυτή την έρευνα.

Μερικές προτάσεις είναι οι εξής:

1. Η εφαρμογή του μοντέλου RNN ή και του ARIMA προσπαθώντας να προβλέψει των αριθμών προσβάσεων σελίδων για περισσότερες από μια εφαρμογή που τρέχουν ταυτόχρονα.
2. Η προσπάθεια εύρεσης χαρακτηριστικών – μοτίβων που ισχύουν για όλες τις τοποθετήσεις σελίδων στις μονάδες αποθήκευσης ανεξάρτητος εφαρμογής σε υβριδικά συστήματα μνήμης. Έρευνα trial – error! Αν επιτευχθεί αυτό θα λυθούν πολλαπλά προβλήματα στον κόσμο βελτιστοποίησης στον τομέα της πληροφορικής αλλά και ευρύτερα.
3. Ανάπτυξη της ίδιας εργασίας με την διαφορά ότι οι χρονοσειρές που αναπτύσσονται θα έχουν περισσότερες κρίσιμες πληροφορίες (πχ. η προσθήκη μιας παραμέτρου χρόνου, των αριθμών προσβάσεων που είχε η σελίδα στην προηγούμενη εποχή κτλ.). Καθώς και ότι οι εποχές δρομολόγησης θα οριστούν με τον αριθμό 1000.
4. Ανάπτυξης ενός δρομολογητή σελίδων όπου θα συνδυάζει τα ARIMA και τα RNNs σαν μια ενιαία λύση. Τα προβλεπόμενα αποτελέσματα των μοντέλων ARIMA θα δίνονται σαν είσοδος στα μοντέλα RNNs με στόχο να αυξηθεί η δυνατότητα πρόβλεψής τους.

Βιβλιογραφία

- [1] Akstinaite, A. (2023). Time series forecasting by combining LSTM RNN with ARIMA method (Doctoral dissertation).
- [2] Doudali, T. D., & Gavrilovska, A. (2022, May). Coeus: Clustering (A) like Patterns for Practical Machine Intelligent Hybrid Memory Management. In 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (pp. 615-624). IEEE.
- [3] Doudali, T. D., Blagodurov, S., Vishnu, A., Gurumurthi, S., & Gavrilovska, A. (2019, June). Kleio: A hybrid memory page scheduler with machine intelligence. In Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing (pp. 37-48).
- [4] Doudali, T. D., Zahka, D., & Gavrilovska, A. (2021). Cori: Dancing to the Right Beat of Periodic Data Movements over Hybrid Memory Systems. 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Portland, OR, USA, pp. 350-359. doi: 10.1109/IPDPS49936.2021.00043.
- [5] Fang, W., Zhu, R., & Lin, J. C. W. (2023). An air quality prediction model based on improved Vanilla LSTM with multichannel input and multiroute output. Expert Systems with Applications, 211, 118422.
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. The MIT Press.
- [7] Graves, A., Jaitly, N., & Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional LSTM. In 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (pp. 273–278).
- [8] Hashemi, M., Swersky, K., Smith, J. A., Ayers, G., Litz, H., Chang, J., Kozyrakis, C., & Ranganathan, P. (2018). Learning memory access patterns.

- [9] Kaur, J., Parmar, K. S., & Singh, S. (2023). Autoregressive models in environmental forecasting time series: a theoretical and application review. *Environmental Science and Pollution Research*, 1-25.
- [10] Latif, N., Selvam, J. D., Kapse, M., Sharma, V., & Mahajan, V. (2023). Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices. *Australasian Accounting, Business and Finance Journal*, 17(1), 256-276.
- [11] Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., & Marr, D. (2016). Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC. 2016 International Conference on Field-Programmable Technology (FPT), Xi'an, China, pp. 77-84. doi: 10.1109/FPT.2016.7929192.
- [12] Pluralsight. (2020, September). Introduction to LSTM units in RNN. <https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn>
- [13] ResearchGate. (n.d.). The standard RNN and unfolded RNN. https://www.researchgate.net/figure/The-standard-RNN-and-unfolded-RNN_fig1_318332317
- [14] Scargall, S. (2020). Programming persistent memory: A comprehensive guide for developers (p. 438). Springer Nature. <https://library.oapen.org/handle/20.500.12657/22836>
- [15] Shen, D., Liu, X., & Lin, F. X. (2016). Characterizing emerging heterogeneous memory. In ISMM 2016. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2926697.2926702>
- [16] Greek Wikipedia. (2023, April). Μηχανική μάθηση. https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE_%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7
- [18] GTkernel. (2022, March). Coeus-sim. <https://github.com/GTkernel/coeus-sim>

- [19] Dulloor, S. R., Roy, A., Zhao, Z., Sundaram, N., Satish, N., Sankaran, R., Jackson, J., & Schwan, K. (2016). Data tiering in heterogeneous memory systems. In Proceedings of the Eleventh European Conference on Computer Systems, EuroSys '16. New York, NY, USA: Association for Computing Machinery.
<https://doi.org/10.1145/2901318.2901344>

Παράρτημα Α

Περισσότερες πληροφορίες που αφοράνε την εκτέλεση των RNNs για όλες τις εφαρμογές:

*Συμβολισμοί για πολιτικές: H - history, HY - hybrid, HOEBF - hybrid-oracle exclusively for 100% of the beneficial pages, HOEASR - hybrid-oracle exclusively for assessing the success rate of RNNs for beneficial pages και O - oracle.

EXPERIMENT 001	NEURONS	LEARNING RATE	TRAINING EPOCH	WINDOW LENGTH	NUMBER OF REQUEST PER SCED. EPOCHS	BENEFITUAL PAGES	SPLIT DATA	DRAM RATION
DETAILS	128	0.001	100	16	CORI	100	0.3	0.2

INFO/TRACE	cpd					hotspot					kmeans				
Fast_Hitrate	60.28	45.91	64.81	61.37	74.8	68.4	64.03	68.45	68.33	69.09	89.69	89.66	91.7	91.03	92.73
Period_Duration	35500	35500	35500	35500	35500	24900	24900	24900	24900	24900	26000	26000	26000	26000	26000
Number_of_Periods	26	26	26	26	26	26	26	26	26	26	20	20	20	20	20
Num_Pages	954	954	954	954	954	422	422	422	422	422	572	572	572	572	572
Num_Pages_Misplaced	422	428	319	397	0	82	82	16	19	0	175	175	32	78	0
Policy*	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O
Num_Pages_Under_Oracle	0	0	100	24	0	0	0	82	72	0	0	0	100	57	0
Num_RNN	0	100	0	0	0	0	82	0	0	0	0	100	0	0	0

INFO/TRACE	lud					pennant					quicksilver				
Fast_Hitrate	59.38	48.48	61.81	59.77	66.96	51.54	34.33	62.84	55.98	84.44	30.94	33.23	36.09	33.6	80.83
Period_Duration	23200	23200	23200	23200	23200	11700	11700	11700	11700	11700	31100	31100	31100	31100	31100
Number_of_Periods	31	31	31	31	31	45	45	45	45	45	36	36	36	36	36
Num_Pages	590	590	590	590	590	890	890	890	890	890	2179	2179	2179	2179	2179
Num_Pages_Misplaced	318	317	211	306	0	448	445	339	391	0	1623	1629	1501	1568	0
Policy*	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O
Num_Pages_Under_Oracle	0	0	100	12	0	0	0	100	49	0	0	0	100	62	0
Num_RNN	0	100	0	0	0	0	100	0	0	0	0	100	0	0	0

INFO/TRACE	backprop					bfs					bptree				
Fast_Hitrate	15.27	15.21	31.04	21.9	76.67	61.14	60.95	63.48	62.01	90.14	56.97	56.57	60.36	59.04	88.89
Period_Duration	9000	9000	9000	9000	9000	12400	12400	12400	12400	12400	27600	27600	27600	27600	27600
Number_of_Periods	68	68	68	68	68	54	54	54	54	54	21	21	21	21	21
Num_Pages	670	670	670	670	670	1560	1560	1560	1560	1560	2208	2208	2208	2208	2208
Num_Pages_Misplaced	466	467	361	420	0	1328	1327	1228	1290	0	1953	1953	1860	1920	0
Policy*	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O
Num_Pages_Under_Oracle	0	0	100	41	0	0	0	100	38	0	0	0	100	33	0
Num_RNN	0	100	0	0	0	0	100	0	0	0	0	100	0	0	0

Παράρτημα Β

Περισσότερες πληροφορίες που αφορούν την εκτέλεση των ARIMA σε όλες τις εφαρμογές:

*Συμβολισμοί για πολιτικές: H - history, HY - hybrid, HOEBF - hybrid-oracle exclusively for 100% of the beneficial pages, HOEASR - hybrid-oracle exclusively for assessing the success rate of RNNs for beneficial pages και O - oracle.

INFO/TRACE	cpd					hotspot					kmeans				
Fast_Hitrate	60.28	45.91	64.81	61.18	74.8	57.24	58.16	62.99	58.35	63.9	90.07	90.07	91.91	90.75	92.78
Period_Duration	35500	35500	35500	35500	35500	20000	20000	20000	20000	20000	24000	24000	24000	24000	24000
Number_of_Periods	26	26	26	26	26	33	33	33	33	33	22	22	22	22	22
Num_Pages	954	954	954	954	954	422	422	422	422	422	572	572	572	572	572
Num_Pages_Misplaced	422	428	319	405	0	146	146	30	125	0	203	203	41	151	0
Policy*	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O
Num_Pages_Under_Oracle	0	0	100	16	0	0	0	100	19	0	0	0	100	34	0
Num_ARIMA	0	100	0	0	0	0	100	0	0	0	0	100	0	0	0

INFO/TRACE	lud					pennant					quicksilver				
Fast_Hitrate	59.38	48.48	61.81	59.77	66.96	51.54	34.33	62.84	53.17	84.44	30.94	33.23	36.09	33.42	80.83
Period_Duration	23200	23200	23200	23200	23200	11700	11700	11700	11700	11700	31100	31100	31100	31100	31100
Number_of_Periods	31	31	31	31	31	45	45	45	45	45	36	36	36	36	36
Num_Pages	590	590	590	590	590	890	890	890	890	890	2179	2179	2179	2179	2179
Num_Pages_Misplaced	318	317	211	306	0	448	445	339	418	0	1623	1629	1501	1578	0
Policy*	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O
Num_Pages_Under_Oracle	0	0	100	12	0	0	0	100	27	0	0	0	100	57	0
Num_ARIMA	0	100	0	0	0	0	100	0	0	0	0	100	0	0	0

INFO/TRACE	backprop					bfs					bptree				
Fast_Hitrate	15.27	15.21	31.04	20.46	76.67	61.14	60.95	63.48	61.76	90.14	56.97	56.57	60.36	58.95	88.89
Period_Duration	9000	9000	9000	9000	9000	12400	12400	12400	12400	12400	27600	27600	27600	27600	27600
Number_of_Periods	68	68	68	68	68	54	54	54	54	54	21	21	21	21	21
Num_Pages	670	670	670	670	670	1560	1560	1560	1560	1560	2208	2208	2208	2208	2208
Num_Pages_Misplaced	466	467	361	432	0	1328	1327	1228	1299	0	1953	1953	1860	1924	0
Policy*	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O	H	HY	HOEBF	HOEASR	O
Num_Pages_Under_Oracle	0	0	100	30	0	0	0	100	29	0	0	0	100	27	0
Num_ARIMA	0	100	0	0	0	0	100	0	0	0	0	100	0	0	0