



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτόματη Συγγραφή Κώδικα με Αναδραστικά  
Νευρωνικά Δίκτυα (Source Code Generation  
with Recurrent Neural Networks)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΒΑΣΙΛΗ ΜΠΟΥΝΤΡΗ

Επιβλέποντες: Ανδρέας Συμεωνίδης, Επίκουρος Καθηγητής Α.Π.Θ.  
Κυριάκος Χατζηδημητρίου, Μεταδιδάκτορας Α.Π.Θ.

ΕΡΓΑΣΤΗΡΙΟ ΕΥΦΥΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ (ISSEL)

Θεσσαλονίκη, Ιούνιος 2017



# Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω τον καθηγητή κ. Ανδρέα Συμεωνίδη για την εμπιστοσύνη και την καθοδήγηση και τον κ. Κυριάκο Χατζηδημητρίου για την καθοδήγηση και τη συνεργασία στα πλαίσια αυτής της διπλωματικής εργασίας. Επίσης θέλω να ευχαριστήσω την οικογένειά μου για την αδιάληπτη στήριξη όλα αυτά τα χρόνια.



# Περίληψη

Η περίληψη θα συμπληρωθεί αργότερα. Αυτή είναι μια περίληψη άλλης εργασίας:

Ένα σύστημα ομότιμων κόμβων αποτελείται από ένα σύνολο αυτόνομων υπολογιστικών κόμβων στο Διαδίκτυο, οι οποίοι συνεργάζονται με σκοπό την ανταλλαγή δεδομένων. Στα συστήματα ομότιμων κόμβων που χρησιμοποιούνται ευρέως σήμερα, η αναζήτηση πληροφορίας γίνεται με χρήση λέξεων κλειδιών. Η ανάγκη για πιο εκφραστικές λειτουργίες, σε συνδυασμό με την ανάπτυξη του Σημασιολογικού Ιστού, οδήγησε στα συστήματα ομότιμων κόμβων βασισμένα σε σχήματα. Στα συστήματα αυτά κάθε κόμβος χρησιμοποιεί ένα σχήμα με βάση το οποίο οργανώνει τα τοπικά διαθέσιμα δεδομένα. Για να είναι δυνατή η αναζήτηση δεδομένων στα συστήματα αυτά υπάρχουν δύο τρόποι. Ο πρώτος είναι όλοι οι κόμβοι να χρησιμοποιούν το ίδιο σχήμα κάτι το οποίο δεν είναι ευέλικτο. Ο δεύτερος τρόπος δίνει την αυτονομία σε κάθε κόμβο να επιλέγει όποιο σχήμα θέλει και απαιτεί την ύπαρξη κανόνων αντιστοίχισης μεταξύ των σχημάτων για να μπορούν να αποτιμώνται οι ερωτήσεις. Αυτός ο τρόπος προσφέρει ευελιξία όμως δεν υποστηρίζει την αυτόματη δημιουργία και τη δυναμική ανανέωση των κανόνων, που είναι απαραίτητες για ένα σύστημα ομότιμων κόμβων.

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος ομότιμων κόμβων βασισμένο σε σχήματα το οποίο (α) θα επιτρέπει μια σχετική ευελιξία στην χρήση των σχημάτων και (β) θα δίνει την δυνατότητα μετασχηματισμού ερωτήσεων χωρίς την ανάγκη διατύπωσης κανόνων αντιστοίχισης μεταξύ σχημάτων, χρησιμοποιώντας κόμβους με σχήματα RDF που αποτελούν υποσύνολα-όψεις ενός βασικού σχήματος (καθολικό σχήμα).

## Λέξεις Κλειδιά

Σύστημα ομότιμων κόμβων, Σύστημα ομότιμων κόμβων βασισμένο σε σχήματα, Σημασιολογικός Ιστός, RDF/S, RQL, Jxta



# Abstract

The Compact Linear Collider (CLIC) will use a novel acceleration scheme in which energy extracted from a very intense beam of relatively low-energy electrons (the Drive Beam) is used to accelerate a lower intensity Main Beam to very high energy. The high intensity of the Drive Beam, with pulses of more than  $10^{15}$  electrons, poses a challenge for conventional profile measurements such as wire scanners. Thus, new non-invasive profile measurements are being investigated.

One candidate is the Electron Beam Scanner. A probe beam of low-energy electrons crosses the accelerator beam perpendicularly. The probe beam is deflected by the space-charge fields of the accelerator beam. By scanning the probe beam and measuring its deflection with respect to its initial position, the transverse profile of the accelerator beam can be reconstructed.

Analytical expressions for the deflection exist in the case of long bunches, where the charge distribution can be considered constant during the measurement. In this paper we consider the performance of an electron beam scanner in an accelerator where the bunch length is much smaller than the probe-beam scanning time. In particular, the case in which the bunch length is shorter than the time taken for a particle of the probe beam to cross the main beam is difficult to model analytically. We have developed a simulation framework allowing this situation to be modelled.

## Keywords

Fill in





# Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	5
Περιεχόμενα	8
Κατάλογος Σχημάτων	9
Κατάλογος Πινάκων	11
<b>1 Εισαγωγή</b>	<b>13</b>
1.1 Κίνητρο	14
1.2 Περιγραφή του προβλήματος	14
1.3 Στόχοι της διπλωματικής	14
1.4 Μεθοδολογία	15
1.5 Διάρθρωση	16
<b>2 Θεωρητικό υπόβαθρο</b>	<b>17</b>
2.1 Deep Learning[3]	17
2.2 Supervised Learning[3]	18
2.3 Recurrent Neural Networks	19
2.4 Εκπαίδευση των Recurrent Neural Networks	21
2.4.1 Long Short-Term Memory Units[5]	21
<b>3 Μέθοδοι προσομοίωσης</b>	<b>23</b>
3.1 Προσομοίωση με το CST Particle Studio	23
3.2 Προσομοίωση με το CST Particle Studio και το MATLAB	23
<b>4 Αποτελέσματα προσομοίωσης</b>	<b>25</b>
<b>5 Επίλογος</b>	<b>27</b>
5.1 Συμπεράσματα	27

5.2 Μελλοντικές Επεκτάσεις . . . . .	27
Βιβλιογραφία	28
Α΄ Μεταφράσεις Ξένων όρων	31
Β΄ Το μοντέλο στο CST Particle Studio	33
Γ΄ Ο κώδικας MATLAB	35

# Κατάλογος Σχημάτων

2.1	Ένα τυπικό δομικό διάγραμμα επιτηρούμενης εκμάθησης. . . . .	19
2.2	Ένα αναδραστικό νευρωνικό δίκτυο είναι ένα πολύ “βαθύ” πλήρως συνδεδεμένο νευρωνικό δίκτυο του οποίου τα βάρη επαναχρησιμοποιούνται στις διάφορες χρονικές στιγμές. Η μη γραμμική συνάρτηση ενεργοποίησης που χρησιμοποιεί η κρυφή κατάσταση είναι η πηγή της πλούσιας δυναμικής του συστήματος. . . .	20



# Κατάλογος Πινάκων



# Κεφάλαιο 1

## Εισαγωγή

Η πράξη του προγραμματισμού, δηλαδή η ανάπτυξη μίας διαδικασίας με στόχο την επίτευξη ενός έργου, είναι μια εντυπωσιακή επίδειξη των δυνατοτήτων συλλογιστικής του ανθρώπινου εγκεφάλου. Η αυτοματοποίηση της συγγραφής κώδικα και προγραμμάτων (Αυτόματος Προγραμματισμός) είναι ένας στόχος με μακρόχρονη ιστορία, τόσο για τους μηχανικούς λογισμικού, όσο και για τον κλάδο της τεχνητής νοημοσύνης. Ο ακριβής ορισμός του “Αυτόματου Προγραμματισμού” παραμένει ένα θέμα στο οποίο υπάρχει ασυμφωνία μεταξύ των ειδικών, γεγονός που ενισχύεται από την συνεχή αλλαγή του όρου χάρη στις εξελίξεις της τεχνολογίας. Ο David Parnas, αναζητώντας την ιστορία του όρου, καταλήγει: “Ο αυτόματος προγραμματισμός ήταν πάντα ένας ευφημισμός για προγραμματισμό σε μια υψηλότερου επιπέδου γλώσσα από αυτή που είναι διαθέσιμη στον προγραμματιστή.” [5]

Δεδομένης της εγγενούς δυσκολίας και πολυπλοκότητας του στόχου υπάρχει πληθώρα προκλήσεων αλλά και προσεγγίσεων στη λύση του. Δύο σημαντικές ομάδες προσεγγίσεων είναι [1], [6]:

### 1. Επαγωγικός Προγραμματισμός (Induction Programming)

Χρησιμοποιώντας τεχνικές τόσο από τον προγραμματισμό όσο και από την τεχνητή νοημοσύνη στοχεύει στη μάθηση προγραμμάτων, τυπικά δηλωτικών και συχνά αναδρομικών. Για την εκμάθηση χρησιμοποιούνται μη αυστηρές προδιαγραφές, όπως παραδείγματα εισόδου - εξόδου ή περιορισμοί.

### 2. Παραγωγή Κώδικα Βάσει Μοντέλων (Model-Driven Code Generation)

Στην προσπάθεια των ερευνητών λογισμικού να απλοποιήσουν την “διαδρομή” ανάμεσα στη σχεδίαση και την υλοποίηση χρησιμοποιούνται αφαιρέσεις. Ένα σύνολο τεχνικών με ευρεία και αυξανόμενη χρήση, το οποίο βασίζεται σε τέτοιες αφαιρέσεις, είναι το Model Driven Engineering. Γλώσσες μοντελοποίησης που αφορούν συγκεκριμένους τομείς (Domain-specific modeling languages) χρησιμοποιούνται σε συνδυασμό με συστήματα μετατροπών και παραγωγής (transformation engines and generators) για να φτιάξουν αντικείμενα όπως κώδικα, προσομοιώσεις εισόδου ή και άλλα μοντέλα.

Με ένα λειτουργικό σύστημα αυτόματης παραγωγής κώδικα ο χρόνος ανάπτυξης και ο

αριθμός των λαθών μειώνεται δραματικά. Αντίστροφα, εκτινάσσεται η παραγωγικότητα των χρηστών και απλουστεύεται η αντιμετώπιση σύνθετων προβλημάτων.

## Κίνητρο

Αφενός, η πρόοδος της τεχνητής νοημοσύνης, και ειδικότερα του κλάδου της υπολογιστικής εκμάθησης (Machine Learning), είναι ραγδαία τα τελευταία χρόνια. Αφετέρου, η εξέλιξη και η ευρεία χρήση του λογισμικού δημιουργεί ανάγκες για αυτοματοποίηση στην παραγωγή του αλλά και γιγαντιαία ποσότητα υλικού το οποίο μπορούμε να χρησιμοποιήσουμε. Έχουμε στη διάθεση μας πληθώρα υλοποιημένων προγραμμάτων, σε πολλές διαφορετικές γλώσσες και μορφές, κάθε δυσκολίας και σκοπού. Οι σχετικές τεχνολογικές και θεωρητικές ανακαλύψεις ανοίγουν νέα μονοπάτια πειραματισμού, καινούρια εργαλεία αναπτύσσονται και δημιουργούνται κίνητρα επανεξέτασης κάποιων προβλημάτων.

Σύμφωνα με τα παραπάνω, και ιδιαίτερα χάρη στις πρόσφατες προόδους της υπολογιστικής εκμάθησης γύρω από την ταξινόμηση και παραγωγή κειμένου [2], [4], καλούμαστε να εξετάσουμε πως και σε τι βαθμό μπορούμε να τις εκμεταλλευτούμε ως μηχανικοί λογισμικού. Τι εφαρμογές μπορούν να προκύψουν για την πρόβλεψη και τη διόρθωση κώδικα; Μέχρι ποιο σημείο μπορούμε να αυτοματοποιήσουμε την παραγωγή του;

## Περιγραφή του προβλήματος

Το πρόβλημα που τίθεται προς λύση είναι η αυτοματοποίηση της παραγωγής κώδικα. Δεδομένων των σύγχρονων μεθόδων και τεχνολογιών, αυτό είναι ζήτημα στο οποίο είναι από εξαιρετικά δύσκολο έως αδύνατο να δωθεί μια γενική λύση, τουλάχιστον για το εγγύς μέλλον. Αντί για μία γενική λύση, μπορούμε να επικεντρωθούμε στις διεργασίες οι οποίες είναι μεν απλές, αλλά επαναλαμβάνονται συχνά και είναι χρονοβόρες. Ίδανικά, θα θέλαμε να αποφύγουμε να καταβάλουμε κόπο για να δημιουργήσουμε κάτι το οποίο ήδη υπάρχει.

## Στόχοι της διπλωματικής

Στόχος της διπλωματικής εργασίας αυτής είναι η δημιουργία ενός τεχνητού νευρωνικού δικτύου με αναδράσεις (artificial recurrent neural network) το οποίο αφού εκπαιδευτεί στην συγγραφή κώδικα σε μία γλώσσα προγραμματισμού της επιλογής μας – διαβάζοντας εκατομμύρια γραμμές κώδικα – θα προσπαθήσει να παράξει κώδικα. Δεδομένου του εκπαιδευτικού χαρακτήρα της διπλωματικής εργασίας, θα εξετάσουμε το πρόβλημα αυτόματης παραγωγής κώδικα από μία πληροφοριακά οδηγούμενη (data-driven) σκοπιά, η οποία επιχειρεί να εκμεταλλευτεί τις εξελίξεις στην επιστήμη της πληροφορίας.

Ο κώδικας αυτός γενικά μπορεί να φτάσει σε ένα από τα παρακάτω επίπεδα:

1. Να “μοιάζει” με κώδικα
2. Να μην έχει συντακτικά λάθη



3. Να μπορεί να μεταγλωτιστεί
4. Να “κάνει κάτι χρήσιμο”

Σε επίπεδο διπλωματικής εργασίας επιζητούμε κώδικα στα επίπεδα τουλάχιστον 1 ή και 2.

## Μεθοδολογία

Θα αντιμετωπίσουμε την παραγωγή κώδικα ως ένα πρόβλημα εκμάθησης ακολουθιών (Sequence Learning), προσέγγιση η οποία βρίσκεται ανάμεσα στον επαγωγικό προγραμματισμό και την παραγωγή κώδικα βάσει μοντέλων. Χρησιμοποιούμε μοντέλα βασισμένα σε αναδραστικά νευρωνικά δίκτυα και ένα σύνολο δεδομένων. Το τελευταίο αποτελείται από έναν μεγάλο αριθμό προγραμμάτων σε μια γλώσσα της επιλογής μας. Σε αυτή την περίπτωση θα χρησιμοποιήσουμε τη γλώσσα θασεριπ, αλλά το μοντέλο μας είναι αγνωστικό στο ποια γλώσσα μαθαίνει. Η μεθοδολογία μπορεί να χωριστεί, αφαιρετικά, σε 3 μέρη:

### 1. Προ-επεξεργασία (Pre-processing)

Δεδομένου ενός μεγάλου όγκου πληροφοριών σε μορφή κώδικα, καλούμαστε να τις επεξεργαστούμε με στόχο την καλύτερη εκμετάλλευση τους από το μοντέλο μας και τελικώς την επίτευξη βέλτιστων αποτελεσμάτων. Αφαιρούμε την πληροφορία που φαίνεται είτε να δυσκολεύει την εκμάθηση του μοντέλου, είτε είναι αδύνατο να ερμηνευτεί από αυτό. Σε μία από τις προτεινόμενες προσεγγίσεις προσθέτουμε πληροφορία για τον κώδικα με σκοπό την αποσαφήνιση των δεδομένων. Η πληροφορία του κώδικα εκφράζεται ως σειρά από στοιχειώδεις χαρακτήρες.

### 2. Εκπαίδευση (Training)

Τα προτεινόμενα μοντέλα, τα οποία είναι σύνθετες δομές αναδραστικών νευρωνικών δικτύων, εκπαιδεύονται βάσει της παραπάνω επεξεργασμένης πληροφορίας. Μετά από το ‘διάβασμα’ μιας σειράς χαρακτήρων καλούνται να προβλέψουν τον επόμενο χαρακτήρα. Οι επιδόσεις εκφράζονται μέσω μιας μετρικής λάθους, την οποία η εκπαιδευτική διαδικασία προσπαθεί να ελαχιστοποιήσει χρησιμοποιώντας γενικευμένες μεθόδους βελτιστοποίησης.

### 3. Παραγωγή Κώδικα (Source Code Generation)

Τα εκπαιδευμένα, πλέον, μοντέλα μπορούν να χρησιμοποιηθούν για την παραγωγή κώδικα. Αρχικοποιούνται με κώδικα της επιλογής μας, ο οποίος επεξεργάζεται όπως και τα δεδομένα στα οποία εκπαιδεύεται. Το μοντέλο παράγει ένα χαρακτήρα σε κάθε πρόβλεψη και χρησιμοποιεί την πρόβλεψη του ως αληθή για να παράξει τον επόμενο χαρακτήρα. Με αυτό τον τρόπο μπορεί να συγγράφει απεριόριστη ποσότητα κώδικα.

## Διάρθρωση

Η εργασία αυτή είναι οργανωμένη σε πέντε κεφάλαια: Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο των βασικών τεχνολογιών που σχετίζονται με τη διπλωματική αυτή. Αρχικά περιγράφονται ..., στη συνέχεια το ... και τέλος ... . Στο κεφάλαιο 3 παρουσιάζονται Στο Κεφάλαιο 4 αρχικά παρουσιάζεται ανάλυση και η σχεδίαση του συστήματος ... .. Τέλος στο Κεφάλαιο 5 δίνονται τα συμπεράσματα, η συνεισφορά αυτής της διπλωματικής εργασίας, καθώς και μελλοντικές επεκτάσεις.

## Κεφάλαιο 2

# Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά οι ...

### Deep Learning[3]

Η υπολογιστική εκμάθηση (Machine Learning) είναι η κινητήριος δύναμη για διάφορες εκφάνσεις της σύγχρονης κοινωνίας: από αναζητήσεις στο διαδίκτυο μέχρι και φιλτράρισμα περιεχομένου σε κοινωνικά δίκτυα και προτάσεις αγορών σε ηλεκτρονικά καταστήματα. Ολοένα συχνότερη και συνηθέστερη γίνεται η εμφάνιση του σε προϊόντα ευρείας κατανάλωσης όπως κάμερες και κινητά τηλέφωνα. Τα συστήματα υπολογιστικής εκμάθησης χρησιμοποιούνται για την αναγνώριση αντικειμένων σε εικόνες, την αυτόματη καταγραφή προφορικού λόγου, την αντιστοίχιση προϊόντων, νέων, δημοσιεύσεων με τις προτιμήσεις χρηστών. Σε όλες αυτές τις εφαρμογές, είναι αυξανόμενη η χρήση ενός σετ τεχνικών που φέρει το όνομα Deep Learning.

Οι συμβατικές τεχνικές υπολογιστικής εκμάθησης είχαν περιορισμένη δυνατότητα χρήσης της ανεπεξέργαστης πληροφορίας. Για δεκαετίες, η σχεδίαση και η υλοποίηση ενός συστήματος αναγνώρισης προτύπων ή υπολογιστικής εκμάθησης, απαιτούσε προσεκτική προσέγγιση και σημαντική εξειδίκευση στον εκάστοτε τομέα. Αυτό επειδή χρειαζόταν η μετατροπή της ανεπεξέργαστης πληροφορίας σε μία κατάλληλη εσωτερική αναπαράσταση, την οποία το υποσύστημα εκμάθησης – συχνότερα ένας ταξινομητής – θα χρησιμοποιούσε για αναγνωρίσει πρότυπα στις διάφορες εισόδους.

Η εκμάθηση αναπαραστάσεων είναι ένα σύνολο μεθόδων που επιτρέπουν σε ένα σύστημα να ανακαλύψει αυτόματα ποιες ακριβώς αναπαραστάσεις της ανεπεξέργαστης πληροφορίας χρειάζεται για να επιτελέσει την αναγνώριση προτύπων ή την ταξινόμηση. Οι μέθοδοι Deep Learning είναι μέθοδοι εκμάθησης αναπαραστάσεων με πολλαπλά επίπεδα αναπαράστασης, που αποτελούνται από την σύνθεση απλών, μη γραμμικών υποσυστημάτων, το καθένα από τα οποία – ξεκινώντας από την ανεπεξέργαστη είσοδο – μετατρέπει την αναπαράσταση της πληροφορίας σε μια λίγο πιο υψηλά αφαιρετική μορφή σε κάθε επίπεδο. Με την χρήση αρκετών τέτοιων μετατροπών το σύστημα μπορεί να μάθει εξαιρετικά σύνθετες λειτουργίες. Για διαδικασίες ταξινόμησης, τα υψηλότερα επίπεδα αναπαράστασης ενισχύουν πτυχές τις εισόδου που είναι

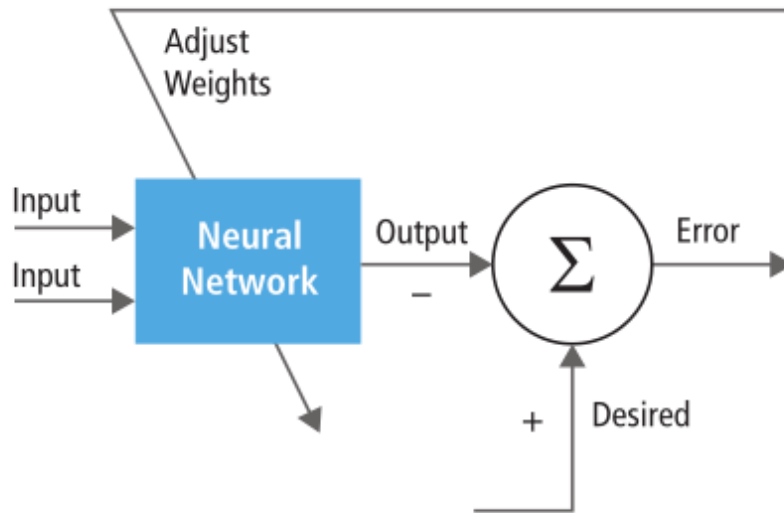
πιο σημαντικές για τον τελικό σκοπό. Σε μία εικόνα, για παράδειγμα, η οποία αναπαριστάται ως διάνυσμα τιμών εικονοκυττάρων, τα χαρακτηριστικά που μαθαίνονται στο πρώτο επίπεδο είναι συνήθως πληροφορία για την παρουσία ή την απουσία ακμών σε συγκεκριμένες θέσεις και προσανατολισμούς. Στο δεύτερο επίπεδο, συνήθως εντοπίζονται μοτίβα μέσω των διαφόρων διατάξεων των ακμών, χωρίς να χρειάζεται τα μοτίβα να επαναλαμβάνονται επακριβώς. Στο τρίτο επίπεδο μπορούν να αναγνωριστούν σύνολα μοτίβων σε μεγάλους συνδυασμούς που αντιστοιχούν σε γνωστά αντικείμενα ή μέρη τους. Τα επόμενα επίπεδα, παρόμοια, εντοπίζουν πιο σύνθετα αντικείμενα ως συνδυασμούς απλούστερων μερών. Το βασικότερο στοιχείο του Deep Learning είναι πως τα επίπεδα που εντοπίζουν χαρακτηριστικά και δομές δεν είναι σχεδιασμένα από τους ανθρώπους: μαθαίνονται από τα δεδομένα χρησιμοποιώντας γενικευμένες διαδικασίες εκμάθησης.

Η χρήση του Deep Learning έχει βοηθήσει στην αντιμετώπιση προβλημάτων που δυσκόλευαν την κοινότητα της τεχνητής νοημοσύνης εδώ και χρόνια. Αποδεικνύεται να έχει επιδόσεις χωρίς προηγούμενο στον εντοπισμό πολύπλοκων δομών σε δεδομένα πολλών διαστάσεων και για αυτό είναι εφαρμόσιμο σε πολλούς διαφορετικούς τομείς, επιστημονικούς, επιχειρησιακούς και κοινωνικοπολιτικούς. Πέρα από επαναστατικές επιδόσεις στην αναγνώριση φωνής και εικόνας, έχει ξεπεράσει άλλες τεχνικές υπολογιστικής εκμάθησης στην πρόβλεψη συμπεριφοράς μορίων φαρμάκων, στην ανάλυση δεδομένων από επιταχυντές σωματιδίων, στην ανακατασκευή εγκεφαλικών κυκλωμάτων και στην πρόβλεψη των επιπτώσεων μεταλλάξεων μη κωδικοποιητικού DNA στις γονιδιακές εκφράσεις και ασθένειες. Ίσως, οι πιο αναπάντεχα υποσχόμενες επιδόσεις έγιναν στον κλάδο της επεξεργασίας φυσικής γλώσσας, συγκεκριμένα στην εντοπισμό θεμάτων, την ανάλυση συναισθήματος, τις ερωτήσεις - απαντήσεις και την μετάφραση.

## Supervised Learning[3]

Η πιο συνήθης μορφή υπολογιστικής εκμάθησης, είτε Deep Learning είτε όχι, είναι αυτή της επιτηρούμενης εκμάθησης. Ας θεωρήσουμε πως θέλουμε να φτιάξουμε ένα σύστημα που αποφασίζει τι περιέχει μια εικόνα, όπως ένα σπίτι, ένα αυτοκίνητο, έναν άνθρωπο ή μία γάτα. Αρχικά, συλλέγουμε ένα αρκετά μεγάλο σύνολο δεδομένων με εικόνες στα οποία σημειώνεται τι αντικείμενο από τα παραπάνω περιέχει κάθε εικόνα. Κατά τη διάρκεια της εκπαίδευσης, δίδουμε στο σύστημα μια εικόνα και αυτό παράγει μία πρόβλεψη, στη μορφή ενός διανύσματος με σκορ για κάθε κατηγορία. Θέλουμε η επιθυμητή κατηγορία να έχει το μεγαλύτερο σκορ πρόβλεψης, αλλά αυτό είναι πολύ δύσκολο πριν την εκπαίδευση. Υπολογίζουμε μία αντικειμενική συνάρτηση με την οποία μετράμε το λάθος (ή την απόσταση) μεταξύ των αποτελεσμάτων του συστήματος και των επιθυμητών αποτελεσμάτων. Το σύστημα, ύστερα, προσαρμόζει τις εσωτερικές του παραμέτρους ώστε να μειώσει το λάθος. Οι εσωτερικές παράμετροι, που συχνότερα στη βιβλιογραφία απαντώνται ως βάρη, είναι πραγματικοί αριθμοί που ορίζουν την λειτουργικότητα εισόδου-εξόδου του συστήματος. Σε ένα τυπικό Deep Learning σύστημα, οι εσωτερικές παράμετροι και τα παραδείγματα που χρησιμοποιούμε για την εκμάθηση του συστήματος μπορεί να είναι εκατοντάδες εκατομμύρια σε αριθμό.

Για την κατάλληλη προσαρμογή των βαρών, ο αλγόριθμος εκμάθησης υπολογίζει ένα διάνυσμα κλίσης, για κάθε βάρος, που δείχνει κατά πόσο και προς ποια κατεύθυνση αλλάζει το λάθος αν αλλάξουμε απειροστά το αντίστοιχο βάρος. Το διάνυσμα των βαρών τελικά ρυθμίζεται έτσι ώστε να έχει αντίθετη φορά με το διάνυσμα κλίσης. Η διαδικασία αυτή είναι μία προσπάθεια ελαχιστοποίησης της συνάρτησης λάθους και μεταγενέστερα της μείωσης, κατά μέσο όρο, των λαθών προβλέψεων του συστήματος.



Σχήμα 2.1: Ένα τυπικό δομικό διάγραμμα επιτηρούμενης εκμάθησης.

Στην πλειοψηφία της σύγχρονης βιβλιογραφίας, και στην παρούσα διπλωματική, ο αλγόριθμος ελαχιστοποίησης που χρησιμοποιείται είναι ο stochastic gradient descent (SGD). Αυτός συνίσταται από την επίδειξη λίγων κάθε φορά, σωστά επισημασμένων, παραδειγμάτων στο σύστημα, τον υπολογισμό των προβλέψεων και του λάθους, τον υπολογισμό του διανύσματος κλίσης και την ρύθμιση των βαρών. Η παραπάνω διαδικασία επαναλαμβάνεται για πολλά μικρά σετ παραδειγμάτων, μέχρι η συνάρτηση στόχου να σταματήσει να μειώνεται. Μετά την εκπαίδευση, οι επιδόσεις του συστήματος μετρώνται σε ένα σύνολο διαφορετικών παραδειγμάτων, έτσι ώστε να εξεταστεί η ικανότητα γενίκευσης του συστήματος σε εισόδους που βλέπει για πρώτη φορά.

## Recurrent Neural Networks

Τα αναδραστικά νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs) είναι μία προσαρμογή των κλασικών, πλήρως συνδεδεμένων νευρωνικών δικτύων, έτσι ώστε τα πρώτα να μπορούν να διαχειριστούν ακολουθίες. Σε κάθε χρονική στιγμή, τα RNNs δέχονται μια είσοδο, ενημερώνουν την εσωτερική τους κατάσταση και παράγουν μία έξοδο. Η πολυδιάστατη εσωτερική κατάσταση, που συχνά απαντάται στη βιβλιογραφία ως κρυφή κατάσταση (hidden state), και η μη γραμμική εξέλιξη της διαχειριζόμενης πληροφορίας δίνουν στα αναδραστικά νευρωνικά δίκτυα μεγάλη εκφραστική ευελιξία και δυνατότητα ενσωμάτωσης και διατήρησης

της πληροφορίας σε μεγάλα χρονικά διαστήματα. Ακόμα και όταν η μη γραμμική συνάρτηση που χρησιμοποιείται από κάθε στοιχείο του PNN είναι εξαιρετικά απλή, η χρήση της σε πολλά επίπεδα και η επανάληψη της σε κάθε χρονική στιγμή οδηγεί σε ένα εξαιρετικά δυναμικό σύστημα.

Τα αναδραστικά νευρωνικά δίκτυα ορίζονται ως εξής: δεδομένης μιας ακολουθίας διανυσμάτων εισόδου  $(x_1, x_2, \dots, x_T)$ , το σύστημα υπολογίζει μία ακολουθία κρυφών καταστάσεων  $(h_1, h_2, \dots, h_T)$  και μία παράγει μια ακολουθία εξόδων  $(o_1, o_2, \dots, o_T)$ , σύμφωνα με τον κάτωθι αλγόριθμο:

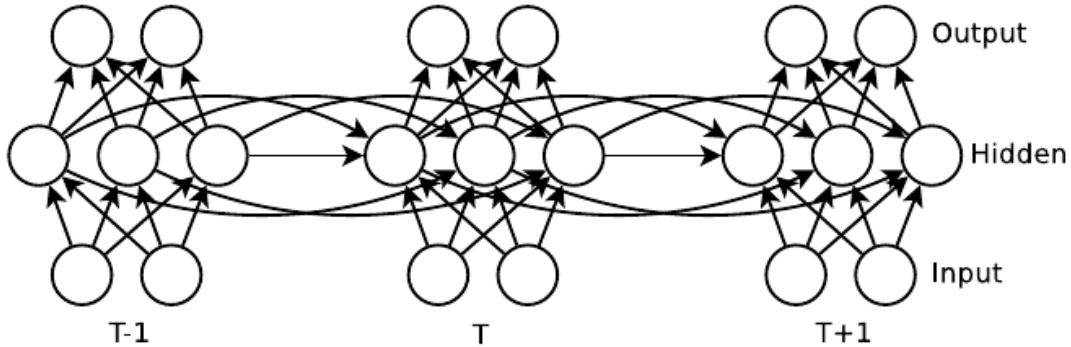
**for**  $t = 1$  to  $T$  **do**

$$h_t = \tanh(W_h x_t + W_{hh} h_{t-1} + b_h) \quad (2.1)$$

$$o_t = W_{oh} h_t + b_o \quad (2.2)$$

**end for**

Σε αυτές τις εξισώσεις, το  $W_{hx}$  είναι ο πίνακας βαρών από την είσοδο στην κρυφή κατάσταση, το  $W_{hh}$  είναι ο πίνακας από την κρυφή κατάσταση στην κρυφή κατάσταση, το  $W_{oh}$  είναι ο πίνακας βαρών από την κρυφή κατάσταση στην έξοδο και τα  $b_h, b_o$  είναι οι σταθεροί όροι. Η μη ορισμένη σχέση  $W_{hh} h_{t-1}$  στη χρονική στιγμή  $t = 1$  αντικαθιστάται με ένα διάνυσμα αρχικοποίησης,  $h_{init}$ , και η συνάρτηση της υπερβολικής εφαπτομένης,  $\tanh$ , εφαρμόζεται κατά στοιχείο.



Σχήμα 2.2: Ένα αναδραστικό νευρωνικό δίκτυο είναι ένα πολύ “βαθύ” πλήρως συνδεδεμένο νευρωνικό δίκτυο του οποίου τα βάρη επαναχρησιμοποιούνται στις διάφορες χρονικές στιγμές. Η μη γραμμική συνάρτηση ενεργοποίησης που χρησιμοποιεί η κρυφή κατάσταση είναι η πηγή της πλούσιας δυναμικής του συστήματος.

Οι παράγωγοι των στοιχειδών μερών του δικτύου είναι εύκολο να υπολογιστούν, με τη μέθοδο της προς τα πίσω διάδοσης σφάλματος, [2], [;] οπότε ίσως η εκπαίδευση ενός τέτοιου συστήματος φαίνεται εύκολη. Στην πραγματικότητα, η σχέση μεταξύ των παραμέτρων του PNN και της δυναμικής του είναι εξαιρετικά ασταθής, γεγονός που καθιστά τον αλγόριθμο SGD αναποτελεσματικό. Αυτό τεκμηριώνεται από [;] και [;] που αποδεικνύουν ότι τα διανύσματα κλίσεων τείνουν να μηδενίζονται (ή σπανιότερα να απειρίζονται) εκθετικά με την διάδοση του σφάλματος στο χρόνο. Στη σχετική βιβλιογραφία αυτό απαντάται ως πρόβλημα

εξαφάνισης ή έκρηξης των διανυσμάτων κλίσης (“vanishing or exploding gradients problem”) Το παραπάνω χρησιμοποιήθηκε ως επιχείρημα για το ότι τα αναδραστικά νευρωνικά δίκτυα δεν μπορούν να αποτυπώσουν εξαρτήσεις με μεγάλη χρονική απόσταση μεταξύ τους, όταν ο χρησιμοποιείται ο αλγόριθμος SGD. Επιπρόσθετα, ο περιστασιακός απειρισμός των διανυσμάτων κλίσης αυξάνει τη διακύμανση τους και κάνει την εκμάθηση ασταθή. Τα θεωρητικά αποτελέσματα αυτά, δεδομένου πως ο SGD ήταν ο βασικότερος αλγόριθμος εκπαίδευσης νευρωνικών δικτύων, σε συνδυασμό με την εμπειρική δυσκολία εκπαίδευσης των RNNs οδήγησε στη σχεδόν ολοκληρωτική εγκατάλειψη της σχετικής έρευνας.

## Εκπαίδευση των Recurrent Neural Networks

### Long Short-Term Memory Units[;]

Ένας τρόπος να αντιμετωπιστεί η αδυναμία που παρουσιάζουν τα PNNs στη δυσκολία εκμάθησης δομών με μακρινές, στο χρόνο, αλληλεξαρτήσεις είναι η τροποποίηση του μοντέλου ώστε να έχει στοιχεία με μνήμη. Η προσέγγιση αυτή ονομάζεται Long Short-Term Memory και γνωρίζει ευρεία χρήση. Οι σχέσεις που ορίζουν κάθε στοιχείο μνήμης είναι:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2.3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2.4)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}h_{t-1} + W_{cf}c_{t-1} + b_c) \quad (2.5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (2.6)$$

$$h_t = o_t \tanh c_t \quad (2.7)$$

όπου το  $\sigma$  είναι η σιγμοειδής συνάρτηση,  $i, f, o$  και  $c$  είναι αντίστοιχα η πύλη εισόδου, η πύλη απώλειας μνήμης, η πύλη εξόδου και η μνήμη. Τα τελευταία είναι διανύσματα με διαστάσεις ίδιες με του διανύσματος  $h$  (βλ. εξίσωση 2.1 που ταυτίζεται με τον δεύτερο όρο της εξίσωσης 2.5).





## Κεφάλαιο 3

# Μέθοδοι προσομοίωσης

Κεφάλαια 3 και 4

- Αποτέλεσμα και σχόλια
- Περιγραφή του CST Particle Studio
- Screenshots

Στο κεφάλαιο αυτό περιγράφεται η υλοποίηση του συστήματος, με βάση τη μελέτη που παρουσιάστηκε στο προηγούμενο κεφάλαιο. Αρχικά παρουσιάζεται η πλατφόρμα και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν. Στη συνέχεια δίνονται οι λεπτομέρειες υλοποίησης για τους βασικούς αλγόριθμους του συστήματος καθώς και η δομή του κώδικα.

## Προσομοίωση με το CST Particle Studio

Το CST Particle Studio μπλα μπλα μπλα.

## Προσομοίωση με το CST Particle Studio και το MATLAB



## Κεφάλαιο 4

# Αποτελέσματα προσομοίωσης

Στο κεφάλαιο αυτό δεν περιγράφεται κάτι (ακόμα)

Some examples illustrating the dependence on bunch intensity, bunch length and transverse size, plus at least on example from the multi-bunch simulations.



## Κεφάλαιο 5

# Επίλογος

### Συμπεράσματα

Συμπεράσματα κλπ

### Μελλοντικές Επεκτάσεις

Το σύστημα που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας θα μπορούσε να βελτιωθεί και να επεκταθεί περαιτέρω, τουλάχιστον ως προς τρεις κατευθύνσεις. Συγκεκριμένα, αναφέρονται τα ακόλουθα:

- Ένα
- Δύο
- Τρία



# Βιβλιογραφία

- [1] Α. Ω. Βιερμανν. Αυτοματις προγραμμιγ : Α τυτοριαλ ον φορμαλ μετηροδολογιες. *Θουρναλ οφ Συμβολις δμπτυατιον*, 1(2):119–142, 1985.
- [2] Α. Γραες. Γενερατιγ Σεχυνεςες ωιτη Ρεσυρρεντ Νευραλ Νετωορκς. *Τεσηνιςαλ Ρεπορτσ*, παγες 1–43, 2013.
- [3] Ψ. Λεϋν, Ψ. Βενγιω, ανδ Γ. Ηιντον. Δεεπ λεαρνιγ. *Νατυρε*, 521(7553):436–444, 2015.
- [4] Π. Λιυ, Ξ. Χιυ, ανδ Ξ. Ηυανγ. Ρεσυρρεντ Νευραλ Νετωορκ φορ Τεξτ “λαςσιφιατιον. 2016.
- [5] Δ. Α. Παρνας. Σοφτωαρε ασπεςτς οφ στρατεγις δεφενσε σψστεμς. *Α“Μ ΣΙΓΣΟΦΤ Σοφτωαρε Εγγινεεριγ Νοτες*, 10(12):15–23, 1985.
- [6] Δ. “. Σζημιδτ. Μοδελ-Δριεν Εγγινεεριγ. *ΙΕΕΕ δμπτυερ*, 39(2):25–31, 2006.





## Παράρτημα Α΄

### Μεταφράσεις Ξένων όρων

#### Μετάφραση

αδερφός  
αμεταβλητότητα  
ανάκτηση πληροφορίας  
αντιμεταθετικότητα  
απόγονος  
απορρόφηση  
βάση δεδομένων  
γνώρισμα  
διαπροσωπεία  
διαφορά  
δικτυακός κατάλογος  
δικτυωτή δομή  
δομικές επερωτήσεις  
δομικές σχέσεις  
δομικό σχήμα  
εγκυρότητα  
ένωση

#### Αγγλικός όρος

sibling  
idempotency  
information retrieval  
commutativity  
descendant  
absorption  
database  
attribute  
interface  
difference  
portal catalog  
lattice  
structural queries  
structural relationships  
schema  
validity  
union



Παράρτημα Β΄

## Το μοντέλο στο CST Particle Studio



Παράρτημα Γ΄

Ο κώδικας MATLAB



