



## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>"Πρόβλεψη τροχιάς Αεροσκάφους με την χρήση Νευρωνικών Δικτύων LSTM"</b> "Prediction of Aircraft trajectory using LSTM Neural Networks"
Όνοματεπώνυμο Φοιτητή	<b>Ζαχόπουλος Γιώργος</b>
Πατρώνυμο	<b>Ανδρέας</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ 14026.....</b>
Επιβλέπων	<b>Νικόλαος Πελέκης, Επίκουρος Καθηγητής</b>

Ημερομηνία Παράδοσης **Ιούνιος 2018**



### **Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Νικόλαος Πελέκης  
Επίκουρος Καθηγητής  
(Επιβλέπων)

Γιάννης Θεοδωρίδης  
Καθηγητής

Άγγελος Πικράκης  
Επίκουρος Καθηγητής



## Περίληψη

Το Long Short-Term Memory (LSTM) είναι μια αρχιτεκτονική ανατροφοδοτούμενου νευρωνικού δικτύου (RNN), το οποίο σχεδιάστηκε για να προσεγγίζει και να μοντελοποιεί χρονικές ακολουθίες και τις μεγάλου εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN. Σε αυτή τη μελέτη, θα χρησιμοποιήσουμε LSTM RNNs για ταξινόμηση τροχιών πτήσεων. Όπως είπαμε, αυτός ο τύπος RNN είναι πολύ αποτελεσματικός για πρόβλεψη και ταξινόμηση ακολουθιών μεγάλου μήκους. Επίσης, θα εκμεταλλευτούμε και το γεγονός ότι τα RNN μπορούν να δεχθούν ως είσοδο ακολουθίες μεταβλητού μήκους. Θα επεξεργαστούμε και θα τροποποιήσουμε τα δεδομένα μας με σκοπό να μπορούμε να τα χρησιμοποιήσουμε στην ανάλυσή μας. Τα δεδομένα θα είναι το γεωγραφικό μήκος, το γεωγραφικό πλάτος και το υψόμετρο, η ταχύτητα, η υγρασία κτλ. Τέλος, θα κατασκευάσουμε το LSTM RNN και να βελτιστοποιήσουμε τις παραμέτρους του, ώστε να πετύχουμε αρκετά μεγάλη ακρίβεια και μετά θα παρουσιάσουμε τα αποτελέσματα.



## **Abstract**

Long Short-Term Memory (LSTM) is a neural network (RNN) feedback architecture designed to approach and model time sequences and their broader dependencies more accurately than other RNN types. In this study, we will use LSTM RNNs to classify flight paths. As we said, this type of RNN is very effective for predicting and classifying length sequences. We will also take advantage of the fact that RNNs can accept sequences of variable length as inputs. We will process and modify the data in order to be able to use it in our analysis. The data will be latitude, latitude and altitude, speed, humidity, etc. Finally, we will build the LSTM RNN and optimize its parameters to achieve fairly high accuracy and then present the results.



## Πίνακας Περιεχομένων

1 – Εισαγωγικά.....	7
2 – Νευρωνικά Δίκτυα .....	8
2.1 – Εισαγωγικά .....	8
2.2 – Μοντέλο του Νευρωνικού Δικτύου.....	8
2.3 – Μαθηματική Περιγραφή .....	11
2.4 – Οι Αισθητήρες Perceptron & MLP .....	11
2.4.1 – Multi-Layer Perceptrons MLPs .....	12
2.5 – Διαδικασίες Μάθησης στα Τεχνητά Νευρωνικά Δίκτυα .....	13
2.5.1 - Μάθηση Πολυεπίπεδων Perceptrons .....	14
2.6 - Recurrent Neural Networks .....	16
2.6.1 - Εκπαίδευση των RNNs .....	21
2.6.2 – Επεκτάσεις των RNNs .....	21
2.6.3 – Εφαρμογές των RNNs .....	23
2.6.4 – BPTT και Vanishing Gradients .....	24
2.6.5 - Το πρόβλημα του Vanishing Gradient .....	26
2.7 - LSTM Neural Networks .....	28
2.7.1 - GRUS Μια Διαφορετική προσέγγιση στα LSTMs .....	30
2.8 - Deep Learning Neural Networks .....	31
2.8.1 - Ορισμός και Ιδιότητες των Βαθιών Αρχιτεκτονικών .....	31
2.8.2 - Restricted Boltzman Machines (RBMs) .....	33
3 – Τεχνικές Πρόβλεψης .....	38
3.1 – Τεχνικές Πρόβλεψης χωρίς χρήση Νευρωνικών Δικτύων .....	38
3.3 – Τεχνικές Πρόβλεψης με Νευρωνικά Δίκτυα .....	44
3.3.1 Σχετικές Εργασίες .....	44
4 – Υλοποίηση – Τεχνικές Πρόβλεψης .....	73
4.1 – Εισαγωγικά .....	73
4.2 – Δεδομένα και Επεξεργασία .....	73



4.3 – Αρχιτεκτονική Δικτύου.....	74
4.3.1 – Εκπαίδευση Δικτύου .....	75
Βιβλιογραφία .....	81



## **Κεφάλαιο 1 – Εισαγωγικά**

Σε αυτή την διπλωματική εργασία θα ασχοληθούμε με την δομή και την λειτουργία των νευρωνικών δικτύων και ειδικότερα με τα LSTM (Long Short Term Memory) νευρωνικά δίκτυα, μια ειδική κατηγορία των RNNs.

Τα RNNs έχουν την δυνατότητα, κατά βάση, να αποθηκεύουν παραστάσεις από πρόσφατα δεδομένα εισόδου (short term memory). Αυτό είναι πολύ σημαντικό για διάφορες εφαρμογές όπως speech recognition, non Markovian Control ακόμα και σύνθεση μουσικής. Οι πιο γνωστοί αλγόριθμοι με τους οποίους υπολογίζεται ποια δεδομένα θα μπουν σαν είσοδος στην short term μνήμη παίρνουν πάρα πολύ χρόνο να εκτελεστούν και αρκετές φορές δεν δουλεύουν καθόλου. Έτσι τις περισσότερες φορές, τα RNNs, δεν καταφέρνουν να έχουν πολύ καλύτερα αποτελέσματα από αντίστοιχες τεχνικές όπως το backpropagation σε feed forward νευρωνικά δίκτυα.

Με τα LSTMs καταφέρνουμε και παρακάμπτουμε τα όποια προβλήματα δημιουργούνται από παλαιότερες αρχιτεκτονικές, επιτυγχάνοντας καλύτερη ακρίβεια στις προβλέψεις μας. Το Long Short-Term Memory (LSTM) είναι μια συγκεκριμένη αρχιτεκτονική ανατροφοδοτούμενου νευρωνικού δικτύου (RNN), το οποίο σχεδιάστηκε για να μοντελοποιεί χρονικές ακολουθίες και τις μεγάλες εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN.

Σε αυτή τη μελέτη, θα χρησιμοποιήσουμε LSTM για πρόβλεψη τροχιών πτήσεων. Όπως είπαμε, αυτός ο τύπος RNN είναι πολύ αποτελεσματικός για πρόβλεψη και ταξινόμηση ακολουθιών μεγάλου μήκους. Με τα LSTM δίκτυα μπορούμε να εκμεταλλευτούμε σε μεγάλο βαθμό τα δεδομένα μακροσκελών ακολουθιών αφού με τα συγκεκριμένα δίκτυα μπορούμε να περιορίσουμε σε μεγάλο βαθμό το πρόβλημα του vanishing gradient, όπως θα δούμε στο κεφάλαιο 2 στην συνέχεια. Τα δεδομένα που θα χρησιμοποιηθούν περιέχουν 693 πτήσεις αεροπλάνων από τη Βαρκελώνη στη Μαδρίτη και 703 πτήσεις από τη Μαδρίτη στη Βαρκελώνη. Σαν είσοδο στον αλγόριθμο που θα χρησιμοποιήσουμε θα τροφοδοτήσουμε όλα τα δεδομένα που μας δίνονται τα οποία είναι:

- Ο χρόνος σε δευτερόλεπτα από την απογείωση
- Η θέση του αεροπλάνου (γεωγραφικό πλάτος και γεωγραφικό μήκος)
- Το υψόμετρο στο οποίο βρίσκεται το αεροπλάνο
- Η θερμοκρασία
- Η ταχύτητα του άνεμου (u, v)
- Η υγρασία

με σκοπό κάθε φορά, βάσει των παραπάνω μεταβλητών, να μπορούμε να προβλέψουμε κάθε επόμενη χρονική στιγμή  $t+1$  την θέση του αεροσκάφους (γεωγραφικό μήκος και γεωγραφικό πλάτος).

Πριν προχωρήσουμε στην πειραματική μελέτη και υλοποίηση, Κεφάλαιο 4, θα ασχοληθούμε, στο Κεφάλαιο 2, με μια γενική περιγραφή στην λειτουργία και εκμάθηση των νευρωνικών δικτύων (Perceptrons και MLPs). Στην συνέχεια του ίδιου κεφαλαίου θα αναλύσουμε εκτενώς τα νευρωνικά δίκτυα RNNs και κατ'επέκταση στα LSTMs δίνοντας έμφαση στις διαδικασίες εκπαίδευσής τους, τους τρόπους χρήσης τους, καθώς και τα μειονεκτήματα και πλεονεκτήματα του κάθε τύπου νευρωνικού δικτύου.

Τέλος στο κεφάλαιο 3 θα παρουσιάσουμε εκτενώς αντίστοιχες εργασίες που ασχολούνται με το πρόβλημα της πρόβλεψης τροχιάς ενός κινούμενου αντικειμένου τόσο με μεθόδους



νευρωνικών δικτύων όσο και με clustering-based και sequential pattern-based μοντέλα πρόβλεψης.

## **Κεφάλαιο 2 - Νευρωνικά Δίκτυα**

Στο κεφάλαιο αυτό θα παρουσιαστούν συνοπτικά οι βασικές αρχές της τεχνητής νοημοσύνης και πιο συγκεκριμένα αυτές των τεχνητών νευρωνικών δικτύων (Neural Networks) καθώς και των νευρωνικών δικτύων τύπου Deep Learning.

### **2.1 Εισαγωγικά**

Η τεχνητή νοημοσύνη έχει σαν σκοπό την ανάπτυξη υπολογιστικών συστημάτων τα οποία θα μιμούνται τα στοιχεία της ανθρώπινης συμπεριφοράς όπως η μάθηση, η προσαρμοστικότητα και η εξαγωγή συμπερασμάτων. Μια από τις κύριες τεχνικές μέσω της οποίας μπορούν να αποδοθούν ανθρώπινες ιδιότητες σε ένα υπολογιστή είναι η χρησιμοποίηση των Τεχνητών Νευρωνικών Δικτύων (Artificial Neural Networks). Παρόλο που οι σύγχρονοι υπολογιστές έχουν πολύ μεγαλύτερη υπολογιστική ισχύ από τον ανθρώπινο εγκέφαλο είναι εξαιρετικά δύσκολο το να προσαρμοστούν με σκοπό να προσομοιάσουν τις λειτουργίες του σε επίπεδο αναγνώρισης μάθησης και προσαρμοστικότητας. Στον ανθρώπινο εγκέφαλο ο κάθε νευρώνας είναι αυτόνομος και ανεξάρτητος και η λειτουργία του είναι ασύγχρονη, δηλαδή δεν απαιτείται συγχρονισμός δεδομένων. Αυτό έχει σαν αποτέλεσμα να αναγνωρίζει τις εικόνες και να επεξεργάζεται τα δεδομένα μέσω της δοκιμής και της επανάληψης, δηλαδή η μάθηση γίνεται μέσω της εμπειρίας. Αντίθετα τα τεχνητά νευρωνικά δίκτυα είναι πολύ δύσκολο να έχουν την ίδια απόδοση με αυτά ενός ανθρώπινου οργανισμού και αυτό οφείλεται κυρίως στην δομή τους. Για να δείξουμε την διαφορά των νευρώνων του εγκεφάλου με τα τεχνητά νευρωνικά δίκτυα, σε τάξεις μεγέθους, αξίζει να αναφέρουμε πως η κατανάλωση ενέργειας στον εγκέφαλο είναι περίπου 20 Watts ενώ ένας υπολογιστής χρειάζεται χιλιάδες...

### **2.2 Το Μοντέλο του Νευρωνικού Δικτύου**

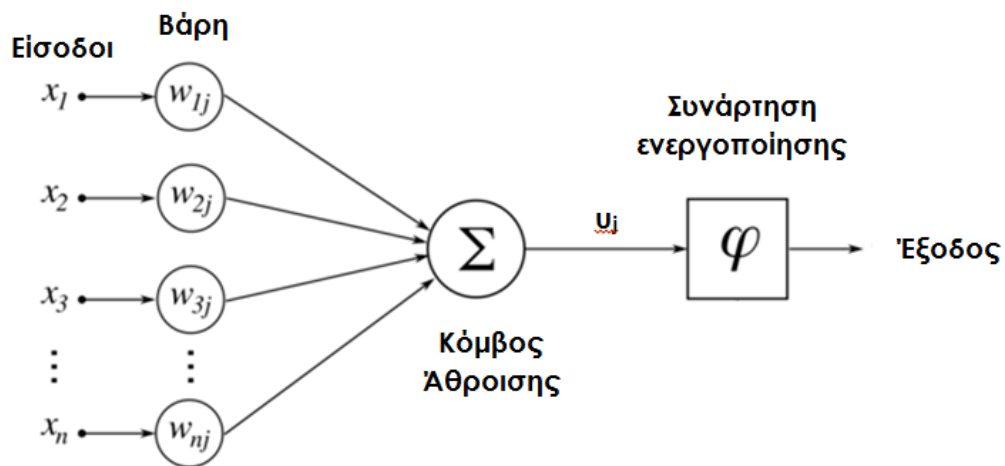
Συνοπτικά ένας νευρώνας είναι μια μονάδα επεξεργασίας πληροφορίας. Τα τρία (3) βασικά στοιχεία του μοντέλου αυτού είναι:

- ✓ Ένα σύνολο από συνάψεις ή συνδετικούς κρίκους.
- ✓ Ένας αθροιστής.
- ✓ Μια συνάρτηση ενεργοποίησης.

Κάθε νευρώνας μπορεί να έχει πολλές εισόδους αλλά μόνο μια έξοδο η οποία μπορεί να αποτελεί και είσοδο για άλλους νευρώνες. Οι συνδέσεις διαφέρουν ως προς τη σημαντικότητα η οποία προσδιορίζεται από έναν συντελεστή βάρους  $w_i$ . Τέλος η επεξεργασία κάθε νευρώνα καθορίζεται από την συνάρτηση μεταφοράς η οποία καθορίζει την κάθε έξοδο ανάλογα με τις εισόδους, και τους συντελεστές βάρους που έχουν υπολογιστεί. Στην συνέχεια παρουσιάζεται μια πιο αναλυτική περιγραφή της διαδικασίας λειτουργίας των νευρώνων,

Σημαντικό στοιχείο για να γνωρίζουμε αποτελεί το γεγονός ότι η πληροφορία στον νευρώνα δεν είναι αμφίδρομη. Η πληροφορία ρέει μόνο προς μία κατεύθυνση (από αριστερά προς τα δεξιά). Στην Εικόνα 3.1 παρουσιάζεται σχηματικά η δομή του.



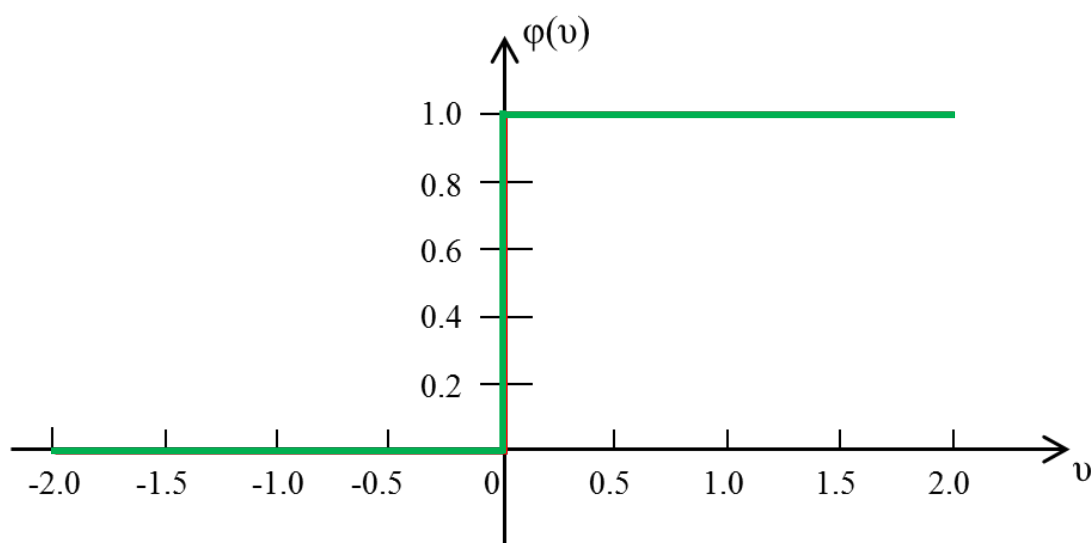


Εικόνα 2.1: Το μοντέλο του νευρωνικού δικτύου

Τα πρώτα μοντέλα νευρώνων, γνωστά και ως μοντέλα McCulloch-Pitts, χρησιμοποιούσαν σαν συνάρτηση ενεργοποίησης μια βηματική συνάρτηση, μια συνάρτηση δηλαδή η οποία μπορούσε να πάρει μόνο τις τιμές 0 ή 1 ανάλογα με την συνθήκη που ικανοποιείται.

$$\varphi(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases}$$

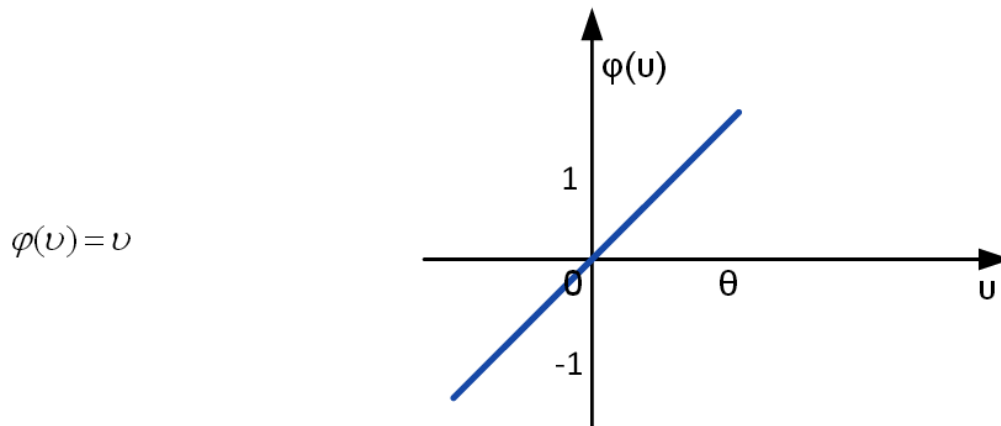
Στην Εικόνα 2.2 παρουσιάζεται γραφικά η εικόνα της συνάρτησης αυτής.



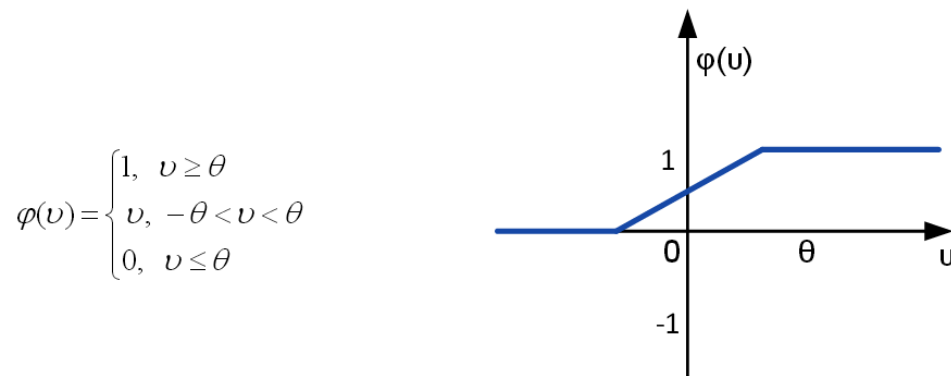
Εικόνα 2.2: Βηματική Συνάρτηση – Συνάρτηση Κατωφλίου



Όπως φαίνεται, από την φύση της βηματικής συνάρτησης, η παράγωγος είναι μηδέν. Μεταγενέστεροι ερευνητές ανακάλυψαν πως θα μπορούσαν να εξάγουν πληροφορία από την παράγωγο της συνάρτησης ενεργοποίησης με αποτέλεσμα να γεννηθεί η ανάγκη για εύρεση μιας νέα συνάρτησης της οποίας η παράγωγος θα ήταν διάφορη από το μηδέν. Γενικά έχουν χρησιμοποιηθεί γραμμικές συναρτήσεις ή τμηματικά γραμμικές συναρτήσεις όπως αυτές που φαίνονται στις Εικόνες 2.3 και 2.4.



Εικόνα 2.3: Γραμμική Συνάρτηση Ενεργοποίησης



Εικόνα 2.4: Τμηματικά Γραμμική Συνάρτηση Ενεργοποίησης

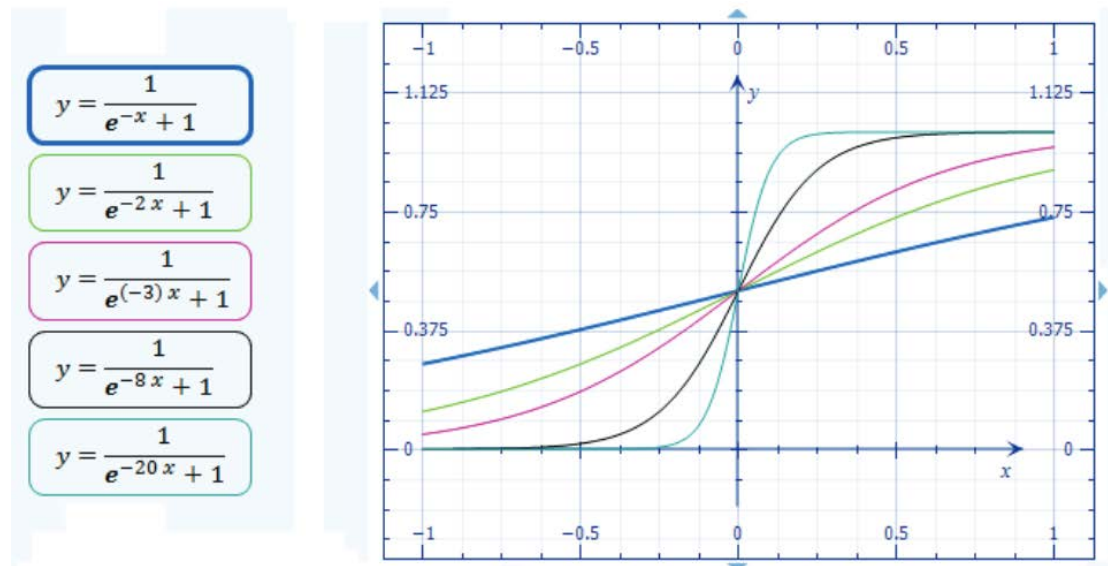
Σήμερα η πλέον χρησιμοποιούμενη συνάρτηση ενεργοποίησης είναι μη γραμμική και έχει την μορφή που παρουσιάζεται στην σχέση

$$\varphi(u) = \frac{1}{1 + e^{-\alpha * u}}$$

όπου  $\alpha$  η παράμετρος κλίσης. Η συνάρτηση αυτή είναι η γνωστή σιγμοειδής συνάρτηση (λόγω της μορφής της γραφικής παράστασης) και ονομάζεται λογιστική συνάρτηση (logistic



function) με πεδίο τιμών (0, 1). Όσο το  $a$  τείνει στο άπειρο η  $\varphi(u)$  τείνει να προσεγγίσει την βηματική συνάρτηση και το μοντέλο McCulloch-Pitts.



Εικόνα 2.5: Σιγμοειδής συνάρτηση για διαφορετικές τιμές του  $a$

### 2.3 Μαθηματική Περιγραφή

Η μαθηματική περιγραφή της λειτουργίας του νευρώνα υπόκειται στις παρακάτω σχέσεις.

$$u_j = \sum_{i=1}^m w_{ij}x_i$$

$$y_i = \varphi(u_j)$$

και

$$u_j = u_j + b_j$$

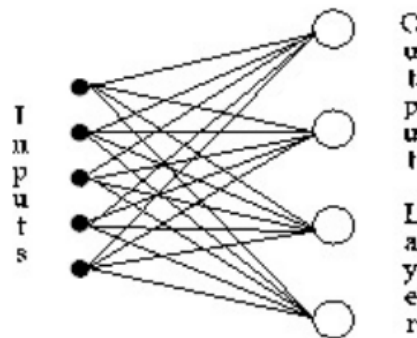
όπου

- $m$  : ο αριθμός των εισόδων που δέχεται ο νευρώνας
- $x_i$  : το  $i$ -οστό σήμα εισόδου του νευρώνα
- $w_{ij}$  : το συναπτικό βάρος του νευρώνα  $j$
- $u_j$  : το σταθμισμένο άθροισμα των εισόδων
- $b_j$  : η πόλωση του νευρώνα
- $u_j$  : το τοπικό πεδίο
- $\varphi(\cdot)$  : η συνάρτηση ενεργοποίησης
- $y_j$  : η έξοδος του νευρώνα



## 2.4 Οι αισθητήρες Perceptron & MLP s

Ο αισθητήρας perceptron είναι ένα δίκτυο με 2 επίπεδα. Το πρώτο επίπεδο απαρτίζεται από τις εισόδους του δικτύου, δεν έχει καθόλου νευρώνες και επομένως δεν γίνεται καμία επεξεργασία πληροφορίας σε αυτό. Το δεύτερο επίπεδο αποτελείται από νευρώνες τύπου McCulloch Pitts και είναι το επίπεδο εξόδου του δικτύου. Ένα παράδειγμα με έξι εισόδους και τέσσερις νευρώνες στο επίπεδο εξόδου φαίνεται στην Εικόνα 3.6.

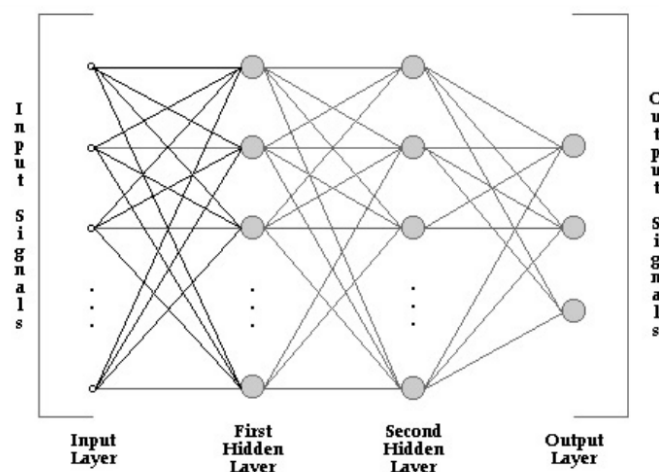


Εικόνα 2.6: Παράδειγμα αισθητήρα perceptron

Ο στόχος του απλού αισθητήρα είναι να μάθει να λύνει προβλήματα ταξινόμησης, να αντιστοιχεί δηλαδή κάθε σετ εισόδων που δέχεται στην σωστή κλάση. Ένα από τα πλεονεκτήματα του δικτύου αυτού είναι ότι υπάρχει ένας σαφής αλγόριθμος βάσει του οποίου μπορεί να εκπαιδευτεί ώστε να δίνει τα σωστά αποτελέσματα. Τα δίκτυα Perceptron χρησιμοποιήθηκαν με επιτυχία σε αρκετές εφαρμογές ταξινόμησης, όμως κατέληξαν στο συμπέρασμα ότι παρουσιάζουν σημαντικά προβλήματα, σε περιπτώσεις όπου οι κλάσεις δεν είναι γραμμικά διαχωρίσιμες. Τα προβλήματα που απλού αυτού μοντέλου ήρθαν να λύσουν τα multi - layer Perceptrons ή αλλιώς MLPs.

### 2.4.1 Multi - Layer Perceptrons MLPs

Οι πολλών στοιβάδων αισθητήρες αποτελούν ένα τροποποιημένο μοντέλο του απλού αισθητήρα perceptron. Σε ένα τέτοιο νευρωνικό δίκτυο, μεταξύ των επιπέδων εισόδου και εξόδου, μεσολαβούν και ένα ή περισσότερα επίπεδα (τα λεγόμενα κρυφά επίπεδα – hidden layers -).





Η ροή της πληροφορίας σε ένα τέτοιο δίκτυο γίνεται πάντα από τα αριστερά προς τα δεξιά. Θεωρούμε επίσης ότι οι νευρώνες σε κάθε επίπεδο επιδρούν με εκείνους τους νευρώνες που ανήκουν στα άμεσα γειτονικά τους επίπεδα. Δηλαδή το πρώτο κρυφό επίπεδο δέχεται μόνο τις τιμές του επιπέδου εισόδου, τα αποτελέσματα του πρώτου κρυφού επιπέδου περνάνε στο δεύτερο κρυφό επίπεδο, του οποίου τα αποτελέσματα περνάνε με την σειρά τους στο επίπεδο εξόδου. Ένα δίκτυο πολυεπίπεδου αισθητήρα στο οποίο υπάρχουν όλες οι δυνατές συνδέσεις μεταξύ τους ονομάζεται πλήρως συνδεδεμένο (fully connected) αλλιώς ονομάζεται μερικώς συνδεδεμένο (partially connected).

Το μοντέλο του κάθε νευρώνα στο δίκτυο περιλαμβάνει μια μη γραμμική συνάρτηση ενεργοποίησης η οποία είναι παντού παραγωγίσιμη (όπως και στα απλά νευρωνικά δίκτυα) η οποία είναι η συνάρτηση που έχει αναφερθεί προηγουμένως.

## **2.5 Διαδικασίες Μάθησης στα Τεχνητά Νευρωνικά Δίκτυα**

Η διαδικασία της μάθησης των νευρωνικών δικτύων είναι μια διαδικασία η οποία δίνει την δυνατότητα στα δίκτυα να μαθαίνουν και να προσαρμόζονται ανάλογα με το περιβάλλον και τα δεδομένα εισαγωγής σε αυτά. Η μάθηση αναφέρεται στη διεργασία επίτευξης μιας επιθυμητής συμπεριφοράς μέσω ανανέωσης των τιμών των συναπτικών βαρών. Έτσι, ένα νευρωνικό δίκτυο μαθαίνει για το περιβάλλον του μέσω μιας επαναληπτικής διαδικασίας ανανέωσης των συναπτικών βαρών και κατωφλίων. Η μάθηση επιτυγχάνεται μέσω αλγορίθμων μάθησης και εκπαίδευσης κάθε ένας από τους οποίους έχει διαφορετικά πλεονεκτήματα και μειονεκτήματα προσφέροντας άλλους τρόπους επιλογής των συναπτικών βαρών. Η πιο ευρέως διαδεδομένη τεχνική μάθησης είναι ο **αλγόριθμος διόρθωσης σφάλματος**, όπου τα βάρη υπολογίζονται βάση των δεδομένων εισόδου και συγκρίνοντας τις προκύπτουσες αποκρίσεις με τις επιθυμητές αποκρίσεις και εν συνεχεία αλλάζοντας τα βάρη προς την κατεύθυνση μείωσης του σφάλματος. Συγκεκριμένα:

$dk(t)$ : Η επιθυμητή τιμή εξόδου του νευρώνα

$x(t)$ : το διάνυσμα εισόδου που εφαρμόζεται στο μοντέλο

$yk(t)$ : Η προβλεπόμενη τιμή της εξόδου του νευρώνα

Το σφάλμα μεταξύ της επιθυμητής τιμής εξόδου  $dk(t)$  και της προβλεπόμενης τιμής εξόδου  $yk(t)$  ορίζεται από την σχέση (3.6) βάσει του οποίου ορίζουμε την συνάρτηση κόστους όπως φαίνεται στην σχέση (3.7).

$$ek(t) = dk(t) - yk(t)$$

$$I = E \left[ \frac{1}{2} \sum_k e_k^2(t) \right]$$

Με τον όρο  $E[...]$  ορίζουμε τον μέσο όρο των τιμών που βρίσκονται μέσα στις αγκύλες. Έτσι η σχέση (3.7) εκφράζει την μέση τιμή του αθροίσματος των τετραγώνων των σφαλμάτων και ονομάζεται κριτήριο Μέσου Τετραγωνικού Σφάλματος (Mean Square Error).

Στην συνέχεια πρέπει να επιλεγούν τα κατάλληλα συναπτικά βάρη έτσι ώστε να ελαχιστοποιηθεί το μέσο τετραγωνικό σφάλμα  $I$ , της οποίας η προσεγγιστική λύση ελαχιστοποιώντας το στιγμιαίο κριτήριο τετραγωνικού σφάλματος είναι:

$$J = \frac{1}{2} \sum_k e_k^2(t)$$



ως προς τα συναπτικά βάρη  $w_{kj}$  του νευρωνικού δικτύου, όπου  $w_{kj}$  είναι το βάρος της σύναψης  $j$  του νευρώνα  $k$ . Ο κανόνας μάθησης Widrow-Hoff ο οποίος βασίζεται στη διόρθωση του σφάλματος είναι τότε:

$$\Delta w_{jk}(t) = \gamma \cdot e_k(t) \cdot x_j(t)$$

όπου:

$\Delta w_{jk}(t)$  : Η ποσότητα αλλαγής στον χρόνο  $t$  του συναπτικού βάρους  $w_{jk}$

$\gamma$  : Μια θετική σταθερά η οποία καθορίζει το ρυθμό μάθησης

Η νέα τιμή  $w_{kj}(t+1)$  του θεωρούμενου συναπτικού βάρους δίνεται από τη σχέση:

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t)$$

### 2.5.1 Μάθηση Πολυεπίπεδων Perceptrons

Η Μηχανική Μάθηση, στην περίπτωση νευρωνικών δικτύων σταθερής δομής, δηλαδή με συγκεκριμένο αριθμό νευρώνων – επιπέδων - συνδέσεων, συνίσταται στον υπολογισμό των ανεξάρτητων παραμέτρων – βαρών των συνδέσεων που αυτά περιέχουν ώστε να εκτελούν μια συγκεκριμένη λειτουργία. Ο πλέον συνήθης αλγόριθμος εκπαίδευσης των πολυεπίπεδων νευρωνικών δικτύων εμπρόσθιας τροφοδότησης είναι αυτός της ανάστροφης διάδοσης σφάλματος (Back Propagation Algorithm). Ο αλγόριθμος αυτός προκύπτει εφαρμόζοντας τις τεχνικές της θεωρίας βελτιστοποίησης πάνω στην αρχιτεκτονική ενός πολυεπίπεδου δικτύου και αποτελεί επέκταση του αλγόριθμου μάθησης των απλών perceptrons όπως αυτός παρουσιάστηκε στην προηγούμενη ενότητα.

Τα 2 βασικά κριτήρια που πρέπει να πληρούνται για τη χρήση αυτής της μεθόδου είναι

α) Το νευρωνικό δίκτυο να περιέχει ένα ή και παραπάνω κρυφά επίπεδα, σε κάθε περίπτωση πάντως όχι μόνο επίπεδα εισόδου και εξόδου (μοντέλο αισθητήρα) και

β) η συνάρτηση ενεργοποίησης να είναι μία μη γραμμική συνάρτηση (συνήθως η σιγμοειδής ή η εφαπτομενική). Σε γενικές γραμμές τα βασικά βήματα που ακολουθούνται για την εκμάθηση ενός ΤΝΔ με αυτή τη μέθοδο είναι τα ακόλουθα :

- Η διαδικασία της μάθησης ξεκινά με τις τιμές των συναπτικών βαρών να θέτονται τυχαία,
- Εφόσον το αποτέλεσμα εξάγει σφάλμα (το πιο πιθανό στην 1η επανάληψη), τα βάρη μετασχηματίζονται ώστε να ελαχιστοποιηθεί το σφάλμα,
- Η διαδικασία αυτή επαναλαμβάνεται κυκλικά πολλές φορές έως ότου το παραγόμενο σφάλμα να μηδενιστεί ή να θεωρηθεί ανεκτό.

Είναι σημαντικό πάντως να επισημανθεί η διακρίσιμότητα των βημάτων σε αυτή τη μέθοδο. Συγκεκριμένα, ο αλγόριθμος οπισθοδιάδοσης του σφάλματος αποτελείται από δύο βασικές λειτουργίες, τη μεταφορά του σήματος προς τα εμπρός (forward pass) και τη μεταφορά του σήματος προς τα πίσω (backward pass).

Στην 1η περίπτωση το σήμα μεταβαίνει από το επίπεδο εισόδου προς το επίπεδο εξόδου μέσω όλων των πιθανών επιπέδων και νευρώνων ανάλογα με τις συνδέσεις. Από αυτή τη μεταφορά σήματος εξάγεται και η όποια απόκριση – έξοδος του δικτύου. Εξαιτίας όμως της διαφοράς, μεταξύ της επιθυμητής και της πραγματικής εξόδου από το δίκτυο, το σήμα



διαδίδεται προς την αντίθετη κατεύθυνση (2η περίπτωση) προκειμένου να τροποποιήσει τα συναπτικά βάρη με στόχο την ελαχιστοποίηση του σφάλματος.

Πιο αναλυτικά παρακάτω παρουσιάζεται σε βήματα η μαθηματική προσέγγιση του αλγόριθμου.

- i. Διαλέγει τα αρχικά βάρη και τις τιμές κατωφλίου χρησιμοποιώντας μικρές θετικές τυχαίες τιμές.
- ii. Παρουσιάζεται στο νευρωνικό δίκτυο το διάνυσμα εκπαίδευσης

$$x(t) = [x_0(t), x_1(t), \dots, x_n(t)]^T$$

και το επιθυμητό διάνυσμα εξόδου.

$$d(t) = [d_1(t), d_2(t), \dots, d_m(t)]^T$$

- iii. Υπολογίζονται τα σήματα εξόδου όλων των νευρώνων του δικτύου προς τα εμπρός χρησιμοποιώντας τις τρέχουσες τιμές των συναπτικών βαρών.

$$y_j(t) = f_j(u_j(t))$$

$$u_j(t) = \sum_i w_{ij}(t)y_i(t)$$

με  $y_i(t)$  να είναι η  $i$  είσοδος του  $j$  νευρώνα (δηλαδή η έξοδος του  $i$  νευρώνα) και  $w_{ij}$  είναι το συναπτικό βάρος που συνδέει τους νευρώνες  $i$  και  $j$ . Για τους νευρώνες της πρώτης κρυφής στοιβάδας η είσοδος και η έξοδος είναι ίδιες. Για τους νευρώνες  $j$  της στοιβάδας εξόδου, το  $y_j(t)$  είναι η  $j$  πραγματική έξοδος του δικτύου.

- iv. Ανανεώνονται τα βάρη ξεκινώντας από τους νευρώνες εξόδου και προς τα πίσω.

$$w_{ij}(t+1) = w_{ij}(t) + \gamma \delta_j(t)y_i(t)$$

όπου:

$$\delta_j(t) = [d_j(t) - y_j(t)] y_j[1 - y_j(t)]$$

η τοπική απόκλιση σε περίπτωση που ο  $j$  νευρώνας ανήκει στην στοιβάδα εξόδου. Σε περίπτωση που ο  $j$  νευρώνας ανήκει σε κρυφή στοιβάδα το  $\delta_j(t)$  δίνεται από τη σχέση:

$$\delta_j(t) = y_j(t)[1 - y_j(t)] \sum_m \delta_m(t)w_{mj}(t)$$

Ο δείκτης  $m$  αφορά σε όλους τους νευρώνες της στοιβάδας εξόδου και το  $w_{ij}$  αναφέρεται στον νευρώνα  $j$  μιας στοιβάδας και στον  $i$  νευρώνα του προηγούμενου επιπέδου.

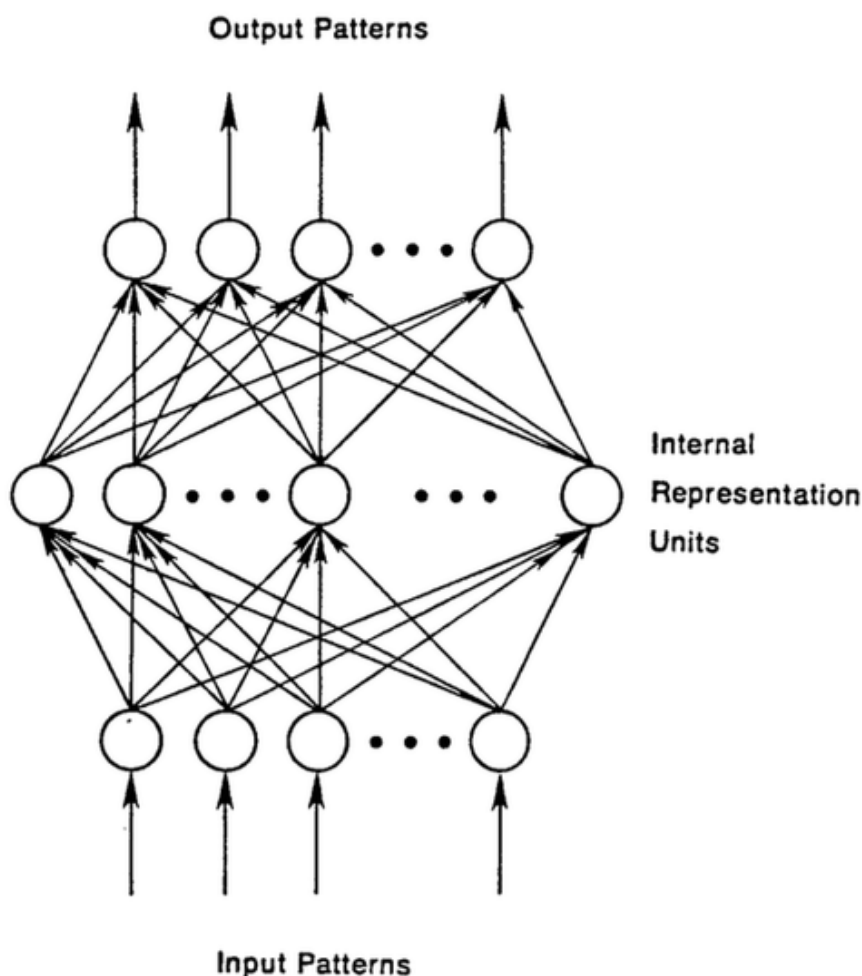
- v. Επαναλαμβάνεται το Βήμα ii.



## 2.6 Recurrent Neural Networks

Αφού έχουμε κατανοήσει τα βασικά των Feed Forward Neural Networks μπορούμε να προχωρήσουμε στην ανάλυση και επεξήγηση των Recurrent Neural Networks (RNNs). Και τα δύο αυτά δίκτυα παίρνουν την ονομασία τους από τον τρόπο που διοχετεύουν τις πληροφορίες τους μέσω μιας σειράς μαθηματικών λειτουργιών που εκτελούνται στους κόμβους του δικτύου. Το ένα τροφοδοτεί τις πληροφορίες κατευθείαν (χωρίς ποτέ να αγγίζει έναν δεδομένο κόμβο δύο φορές), το άλλο τον μετακινεί μέσω βρόχου.

Στην περίπτωση των δικτύων Feed Forward (FNNs), τα παραδείγματα εισόδου τροφοδοτούνται στο δίκτυο και μετατρέπονται σε έξοδο. Με εποπτευόμενη μάθηση (supervised learning), η παραγωγή θα είναι μια ετικέτα. Δηλαδή, κατατάσσουν τα ακατέργαστα δεδομένα σε κατηγορίες, αναγνωρίζοντας πρότυπα που σηματοδοτούν, για παράδειγμα, ότι μια εικόνα εισόδου πρέπει να φέρει την ένδειξη "γάτα" ή "ελέφαντας".



Εικόνα 2.8: Feed Forward Neural Network

Το απλό επαναλαμβανόμενο δίκτυο (Simple Recurrent Neural Network SRN) σχεδιάστηκε και χρησιμοποιήθηκε για πρώτη φορά από τον Jeff Elman και δημοσιεύθηκε για πρώτη φορά σε ένα έγγραφο με τίτλο Finding Structure in Time (Elman, 1990). Το βιβλίο ήταν πρωτοποριακό για πολλούς γνωστικούς επιστήμονες και ψυχολόγους, αφού ήταν ο πρώτος που απομακρύνθηκε πλήρως από μια προηγούμενη δέσμευση σε συγκεκριμένες γλωσσικές



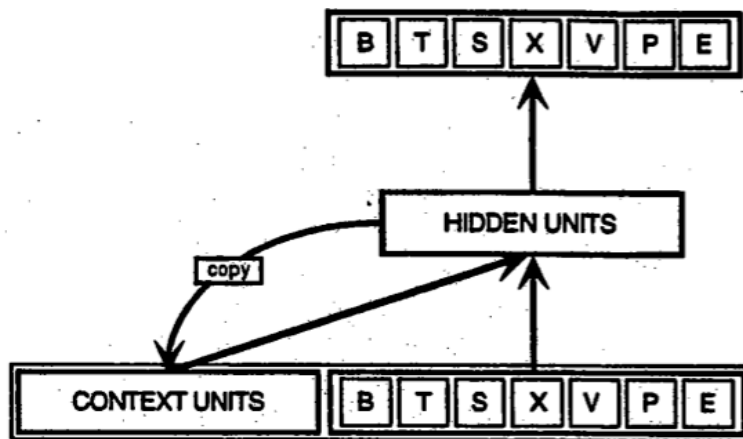


μονάδες (π.χ. φωνήματα ή λέξεις) και κατάφερε να διερευνήσει το όραμα ότι αυτές οι μονάδες μπορεί να είναι οι επερχόμενες συνέπειες μιας μαθησιακής διαδικασίας που λειτουργεί πάνω από την λανθάνουσα δομή στη ροή της ομιλίας. Ο Elman είχε εφαρμόσει ένα προηγούμενο μοντέλο στο οποίο η είσοδος και η έξοδος του δικτύου ήταν μια πολύ χαμηλού επιπέδου φασματογράφηματικού τύπου, και το εκπαίδευσε χρησιμοποιώντας μια φασματική πληροφορία που εξάγεται από μια εγγραφή της δικής του φωνής λέγοντας «Αυτή είναι η φωνή του νευρικού δικτύου». Δεν θα συζητήσουμε τις λεπτομέρειες αυτού του δικτύου, εκτός από το ότι έμαθε να παράγει αυτή την έκφραση μετά από επαναλαμβανόμενη.

Στο μεταγενέστερο έργο του ο Elman επέστρεψε λίγο από την προσέγγιση του ωμού ερεθίσματος που χρησιμοποιήθηκε σε αυτή την αρχική αδημοσίευτη προσομοίωση, αλλά διατήρησε τη θεμελιώδη δέσμευση για την αντίληψη ότι η πραγματική δομή δεν είναι στα σύμβολα που χρησιμοποιούμε ως ερευνητές αλλά στην ίδια την ροή εισόδου. Στην εύρεση της δομής με την πάροδο του χρόνου, ο Elman παρουσίασε διάφορες προσομοιώσεις, μία που αφορούσε την εμφάνιση λέξεων από ένα ρεύμα υπολεξικών στοιχείων (χρησιμοποίησε πραγματικά τα γράμματα που συνθέτουν τις λέξεις ως στοιχεία για αυτό) και το άλλο απευθύνεται στην εμφάνιση προτάσεων Από ένα ρεύμα λέξεων. Και στα δύο μοντέλα, η είσοδος σε οποιαδήποτε χρονική στιγμή προέρχεται από ένα μικρό σταθερό αλφάβητο. Το ενδιαφέρον επικεντρώνεται σε αυτό που μπορεί να μάθει σε μια πολύ απλή αρχιτεκτονική δικτύου, στην οποία το καθήκον που τίθεται στο δίκτυο είναι να προβλέψει το επόμενο στοιχείο στην λέξη, χρησιμοποιώντας το στοιχείο κατά την ώρα  $t$ , καθώς και μια εσωτερική αναπαράσταση της κατάστασης ενός συνόλου από Κρυφές μονάδες (hidden units) από το προηγούμενο χρονικό βήμα  $t-1$ .

Ένα SRN ίδιου τύπου με αυτό που χρησιμοποίησε ο Elman απεικονίζεται στο Σχήμα 2.9. Στην πραγματικότητα παρουσιάζουμε το δίκτυο που χρησιμοποιήθηκε σε μια πρώιμη μελέτη παρακολούθησης από τους Servan-Schreiber et al. (1991), στην οποία χρησιμοποιείται ένα πολύ μικρό αλφάβητο.

Η ομορφιά του SRN είναι η απλότητα του. Στην πραγματικότητα, είναι πραγματικά ένα μόνο δίκτυο τριών επιπέδων, πολλαπλής προώθησης (Feed-Forward Back Propagation Network). Η μόνη προϋπόθεση είναι ότι ένα από τα δύο μέρη της εισόδου στο δίκτυο είναι το μοτίβο ενεργοποίησης πάνω από τις κρυφές μονάδες του δικτύου στο προηγούμενο χρονικό βήμα.

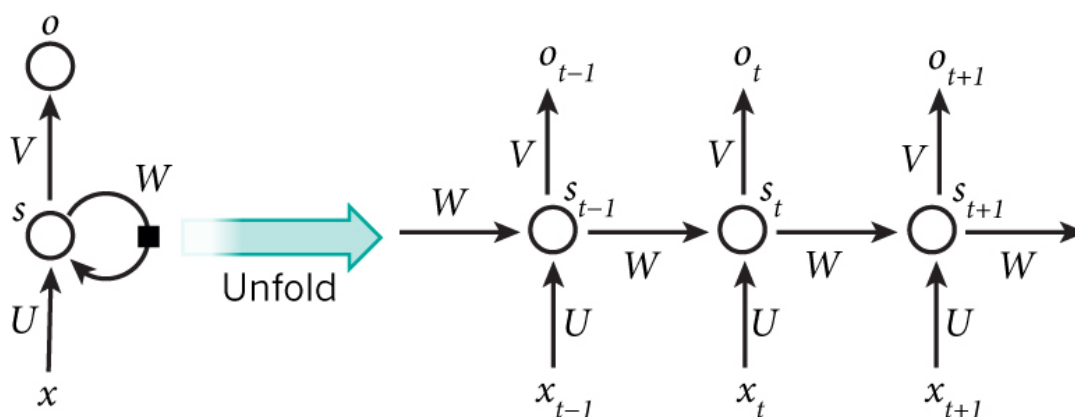


Εικόνα 2.9: Η αρχιτεκτονική του δικτύου SRN. Κάθε κιβώτιο αντιπροσωπεύει μια ομάδα μονάδων και κάθε βέλος προς τα εμπρός αντιπροσωπεύει ένα πλήρες σύνολο εκπαιδευτικών συνδέσεων από κάθε μονάδα αποστολής σε κάθε μονάδα λήψης στην επόμενη ομάδα. Το βέλος προς τα πίσω, από το κρυμμένο στρώμα στο επίπεδο περιεχομένου, υποδηλώνει μια λειτουργία αντιγραφής (Εικόνα από: <https://web.stanford.edu/group/pdplab/pdphandbook/handbookch8.html>).



Πιο αναλυτικά, Τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) είναι δημοφιλή μοντέλα που έχουν δείξει μεγάλη υπόσχεση σε πολλές εργασίες NLP.

Η ιδέα πίσω από τα RNNs είναι ότι κάνουν χρήση διαδοχικών πληροφοριών. Σε ένα παραδοσιακό νευρωνικό δίκτυο υποθέτουμε ότι όλες οι εισοδοι (και οι έξοδοι) είναι ανεξάρτητες η μια από την άλλη. Αυτό για πολλά προβλήματα είναι μια πολύ κακή ιδέα. Εάν θέλουμε να προβλέψουμε την επόμενη λέξη σε μια πρόταση, ξέρουμε καλύτερα ποιες λέξεις ήρθαν μπροστά από αυτήν που προσπαθούμε να προβλέψουμε. Τα RNN ονομάζονται επαναλαμβανόμενα επειδή εκτελούν την ίδια εργασία για κάθε στοιχείο μιας ακολουθίας, με την έξοδο να εξαρτάται από τους προηγούμενους υπολογισμούς. Ένας άλλος τρόπος να σκεφτούμε τα RNNs είναι ότι έχουν μια «μνήμη» που συλλαμβάνει πληροφορίες για το τι έχει υπολογιστεί μέχρι τώρα. Θεωρητικά, τα RNNs μπορούν να χρησιμοποιήσουν πληροφορίες απο αυθαίρετα μακρές αλληλουχίες, αλλά στην πράξη περιορίζονται στην επιστροφή μόνο μερικών βημάτων. Στο παρακάτω σχήμα παρουσιάζεται ένα τυπικό RNN:



**Εικόνα 2.10:** Ένα Recurrent Neural Network και η εξέλιξη του στο χρόνο.

(Εικόνα από: [www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/](http://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/))

Το παραπάνω διάγραμμα δείχνει ένα RNN που ξετυλίγεται (ή ξεδιπλώνεται) σε ένα πλήρες δίκτυο. Με την απελευθέρωση/ ξετύλιγμα απλά εννοούμε ότι γράφουμε το δίκτυο σαν μια πλήρη ακολουθία. Για παράδειγμα, αν η ακολουθία που μας ενδιαφέρει είναι μια πρόταση των 5 λέξεων, το δίκτυο θα ξεδιπλωθεί σε ένα νευρωνικό δίκτυο 5-στρωμάτων, ένα στρώμα για κάθε λέξη. Οι τύποι που διέπουν την λειτουργία και τους υπολογισμούς σε ένα RNN είναι οι εξής:

- $x_t$  είναι η είσοδος για την χρονική στιγμή  $t$  (time step). Για παράδειγμα  $x_1$  θα μπορούσε να είναι ένα άνωσμο το οποίο αντιπροσωπεύει ή αναφέρεται στην δεύτερη λέξη μίας πρότασης.
- $s_t$  είναι η κρυφή κατάσταση την χρονική στιγμή  $t$ . Είναι ουσιαστικά η “μνήμη” του δικτύου που αναφέρθηκε σε προηγούμενη παράγραφο. Το  $s_t$  υπολογίζεται βάσει της προηγούμενης κρυφής κατάστασης και της εισόδου στην τρέχουσα χρονική στιγμή.  $s_t = f(U x_t + W s_{t-1})$ . Η συνάρτηση  $f$  είναι συνήθως μη γραμμική όπως η  $\tanh$  ή  $\text{ReLU}$ . Το  $s_{t-1}$  που απαιτείται για τον υπολογισμό της πρώτης κρυφής κατάστασης, συνήθως αρχικοποιείται σε όλα τα μηδενικά.
- $o_t$  είναι η έξοδος για την χρονική στιγμή  $t$ . Για παράδειγμα εάν θέλαμε να προβλέψουμε την επόμενη λέξη σε μια πρόταση το  $o_t$  θα ήταν ένα διάνυσμα από πιθανότητες κατά μήκος όλου του λεξιλογίου μας.  
 $o_t = \text{softmax}(V s_t)$ .

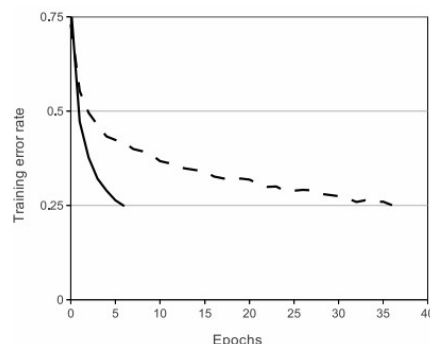
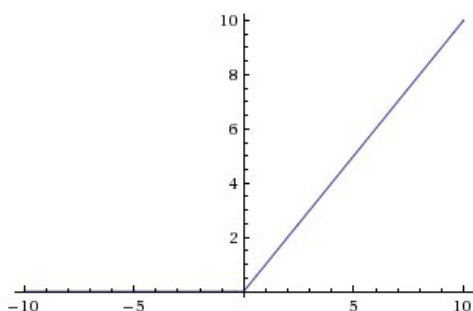


Πριν προχωρήσουμε παρακάτω θα παρουσιάσουμε τις δύο επικρατέστερες συναρτήσεις ενεργοποίησης που αναφέρθηκαν στο βήμα 2.

### ReLU

Η Rectified Linear Unit (ReLU) έχει γίνει πολύ δημοφιλής τα τελευταία χρόνια. Υπολογίζει τη συνάρτηση  $f(x) = \max(0, x)$ . Με άλλα λόγια, η ενεργοποίηση είναι απλώς κατώφλι στο μηδέν. Μια άλλη παραλλαγή είναι η συνάρτηση softplus function η οποία δίνεται από τον τύπο  $\log(1+e^x)$ . Υπάρχουν πολλά πλεονεκτήματα και μειονεκτήματα για τη χρήση των ReLUs:

- ✓ (+) Βρέθηκε ότι επιταχύνει πολύ (π. Χ. κατά ένα συντελεστή 6 σύμφωνα με τους Krizhevsky et al.) τη σύγκλιση της κατάβασης της στοχαστικής κλίσης σε σύγκριση με την σιγμοειδή συνάρτηση (tanh). Υποστηρίζεται ότι αυτό οφείλεται στη γραμμική, μη κορεσμένη μορφή της.
- ✓ (+) Σε σύγκριση με την συνάρτηση ενεργοποίησης tanh / σιγμοειδής συνάρτηση στους νευρώνες που απαιτούνται δαπανηρές ενεργειακά λειτουργίες όπως εκθετικές συναρτήσεις κτλ η ReLU μπορεί να υλοποιηθεί απλά με κατώτατο όριο μιας μήτρας ενεργοποιήσεων στο μηδέν.
- ✓ (-) Δυστυχώς, οι μονάδες ReLU μπορεί να είναι εύθραυστες κατά τη διάρκεια της εκπαίδευσης και μπορούν να "πεθάνουν". Για παράδειγμα, μια μεγάλη κλίση που περνάει μέσα από ένα νευρώνα ReLU θα μπορούσε να αναγκάσει τα βάρη να ενημερωθούν με τέτοιο τρόπο ώστε ο νευρώνας να μην ενεργοποιηθεί ποτέ ξανά σε οποιοδήποτε σημείο των δεδομένων. Εάν συμβεί αυτό, τότε η κλίση που ρέει μέσα από τη μονάδα θα είναι για πάντα μηδέν από εκείνο το σημείο. Δηλαδή, οι μονάδες ReLU μπορούν να "πεθάνουν" και να καταστραφούν χωρίς γυρισμό κατά τη διάρκεια της εκπαίδευσης, καθώς μπορούν να χτυπηθούν από την πολλαπλότητα των δεδομένων. Για παράδειγμα, μπορεί να διαπιστώσετε ότι έως και 40% του δικτύου σας μπορεί να είναι "νεκρό" (δηλαδή νευρώνες που δεν ενεργοποιούνται ποτέ σε ολόκληρο το σύνολο δεδομένων εκπαίδευσης), εάν ο ρυθμός εκμάθησης είναι πολύ υψηλός. Με μια σωστή ρύθμιση του ποσοστού μάθησης αυτό είναι λιγότερο συχνό αυτό το ζήτημα.



**Εικόνα 2.11:** Αριστερά: Λειτουργία ενεργοποίησης γραμμικής μονάδας (ReLU), η οποία είναι μηδενική όταν  $x < 0$  και στη συνέχεια γραμμική με κλίση 1 όταν  $x > 0$   
 Δεξιά: Η γραφική παράσταση από την δημοσίευση των Krizhevsky et al. που δείχνει τη βελτίωση κατά 6 φορές στη σύγκλιση της ReLU σε σύγκριση με την tanh.  
 (Εικόνα από: <http://cs231n.github.io/neural-networks-1/>)



### Σιγμοειδής Συνάρτηση (tanh)

Η σιγμοειδής μη-γραμμική συνάρτηση έχει τη μαθηματική μορφή  $\sigma(x) = 1 / (1 + e^{-x})$ . Όπως αναφέρθηκε στην προηγούμενη ενότητα, παίρνει έναν πραγματικό αριθμό και το "ορίζει" σε εύρος μεταξύ του 0 και του 1. Συγκεκριμένα, οι μεγάλοι αρνητικοί αριθμοί γίνονται 0 και οι μεγάλοι θετικοί αριθμοί γίνονται 1. Η λειτουργία της σιγμοειδούς συνάρτησης έχει ιστορικά συχνή χρήση δεδομένου ότι έχει μια καλή προσέγγιση και εφαρμογή στην λειτουργία έναρξης/ πυροδότησης ενός νευρώνα: από το να μην τον ενεργοποιήσει καθόλου (0) έως πλήρη κορεσμένη ενεργοποίησή του σε μια υποτιθέμενη μέγιστη συχνότητα (1). Στην πράξη, η σιγμοειδής μη-γραμμική συνάρτηση σπάνια χρησιμοποιείται λόγω των δύο βασικών μειονεκτημάτων που έχει:

- ✓ (-) Η σιγμοειδής συναρτήσεις έχουν την τάση να "σκοτώνουν" την κλίση. Μια πολύ ανεπιθύμητη ιδιότητα του νευρώνα που ενεργοποιείται από αυτή την συνάρτηση είναι ότι όταν η ενεργοποίηση του νευρώνα κορεσθεί σε ουρά 0 ή 1, η κλίση στις περιοχές αυτές είναι σχεδόν μηδενική. Ας θυμηθούμε ότι κατά τη διάρκεια του backpropagation, αυτή η (τοπική) κλίση θα πολλαπλασιαστεί με την κλίση της εξόδου αυτής της πύλης για ολόκληρή την έξοδο. Επομένως, εάν η τοπική κλίση είναι πολύ μικρή, θα "σκοτώσει" ουσιαστικά την κλίση και σχεδόν κανένα σήμα δεν θα ρέει μέσα από τον νευρώνα στα βάρη του και αναδρομικά στα δεδομένα του. Επιπλέον, κάποιος πρέπει να δώσει ιδιαίτερη προσοχή κατά την προετοιμασία των βαρών των σιγμοειδών νευρώνων για να αποτρέψει τον κορεσμό. Για παράδειγμα, εάν τα αρχικά βάρη είναι υπερβολικά μεγάλα τότε οι περισσότεροι νευρώνες θα κορεσθούν και το δίκτυο δεν θα εκπαιδευτεί σωστά.
- ✓ (-) Οι έξοδοι ενός τέτοιου δικτύου δεν είναι zero-centered. Αυτό είναι ανεπιθύμητο αφού οι νευρώνες σε μεταγενέστερα στρώματα επεξεργασίας σε ένα Νευρικό Δίκτυο θα λαμβάνουν δεδομένα που δεν είναι μηδενικά κεντραρισμένα. Αυτό έχει συνέπειες στη δυναμική κατά τη διάρκεια της κλίσης κλίσης, επειδή αν τα δεδομένα που εισέρχονται σε ένα νευρώνα είναι πάντα θετικά (π.χ.  $x > 0$ ), τότε η κλίση στα βάρη  $w$  Κατά τη διάρκεια του backpropagation καθίσταται είτε σε όλα θετική είτε σε όλα αρνητική. Αυτό θα μπορούσε να εισαγάγει μια ανεπιθύμητη δυναμική τύπου ζιγκ-ζαγκ στις ενημερώσεις της κλίσης για τα βάρη. Ωστόσο, είναι σημαντικό να παρατηρήσουμε ότι μόλις αυτές οι κλίσεις προστεθούν σε ένα κομμάτι των δεδομένων, η τελική ενημέρωση για τα βάρη μπορεί να έχει μεταβλητές ενδείξεις, μετριάζοντας κάπως αυτό το πρόβλημα. Επομένως, αυτό είναι μια ενόχληση, αλλά έχει λιγότερο σοβαρές συνέπειες σε σύγκριση με το πρόβλημα κορεσμού ενεργοποίησης που αναλύσαμε παραπάνω.

Σε αντίθεση με την σιγμοειδή συνάρτηση η μη γραμμική συνάρτηση tanh είναι zero centered (κεντραρισμένη γύρω από το 0) πράγμα που την καθιστά προτιμότερη από την σιγμοειδή όσον αφορά την ενεργοποίηση ενός νευρώνα.



**Εικόνα 2.12:** Αριστερά: Η σιγμοειδής μη γραμμική συνάρτηση συμπίπτει τους πραγματικούς αριθμούς στο διάστημα μεταξύ του 0 και του 1.

Δεξιά: Η tanh συμπίπτει τους πραγματικούς αριθμούς στο διάστημα μεταξύ του 1 και του -1. (Εικόνα από: <http://cs231n.github.io/neural-networks-1/>)



Κάποιες βασικές αρχές που διέπουν τα RNNs παρουσιάζονται παρακάτω:

- Όπως αναφέρθηκε και προηγουμένως η κρυφή κατάσταση  $s_t$  μπορεί να θεωρηθεί ως η μνήμη του νευρωνικού δικτύου. Κρατάει την πληροφορία για το τι συνέβη σε όλα τα προηγούμενα χρονικά βήματα (timesteps). Πρακτικά η έξοδος  $o_t$  υπολογίζεται μόνο από την μνήμη την χρονική στιγμή  $t$  επειδή το  $s_t$  δεν μπορεί να κρατήσει πληροφορία για πολλές χρονικές στιγμές πίσω.
- Σε αντίθεση με ένα παραδοσιακό deep learning Neural network, το οποίο χρησιμοποιεί διαφορετικές παραμέτρους σε κάθε layer, ένα RNN μοιράζεται τις ίδιες παραμέτρους ( $U$ ,  $V$ ,  $W$  όπως φαίνονται παραπάνω) σε όλα τα βήματα. Αυτό αντανακλά το γεγονός ότι εκτελούμε την ίδια ενέργεια σε κάθε βήμα, μόνο με διαφορετικές εισόδους. Αυτό μειώνει σημαντικά τον συνολικό αριθμό των παραμέτρων που πρέπει να μάθει.
- Όπως φαίνεται στην εικόνα 2.10 το RNN έχει διαφορετικές εξόδους για κάθε χρονική στιγμή. Αυτό όμως δεν είναι ο κανόνας αφού μπορεί να μην χρειαστεί να γίνεται πάντα έτσι. Για παράδειγμα όταν θέλουμε να προβλέψουμε την συνέχεια μια πρότασης δεν χρειάζεται να υπολογίσουμε και να πάρουμε σαν έξοδο όλες τις ακολουθίες των λέξεων αλλά μόνο την τελική πρόταση.

### **2.6.1 Εκπαίδευση των RNNs**

Η εκπαίδευση ενός RNN είναι παρόμοια με την εκπαίδευση ενός κλασικού νευρικού δικτύου. Χρησιμοποιούμε επίσης την τεχνική backpropagation, αλλά με μια μικρή διαφοροποίηση. Επειδή οι παράμετροι μοιράζονται από όλα τα time steps στο δίκτυο, η κλίση σε κάθε έξοδο εξαρτάται όχι μόνο από τους υπολογισμούς για το τρέχον χρονικό σημείο, αλλά και από τα προηγούμενα χρονικά σημεία. Για παράδειγμα, για να υπολογίσουμε την κλίση στο  $t = 4$  θα χρειαζόταν να εφαρμοστεί η τεχνική backpropagate για 3 βήματα και να αθροίσουμε τις κλίσεις. Αυτό ονομάζεται Backpropagation Through Time (BPTT). Το πρόβλημα που πρέπει να σημειωθεί είναι απλώς το γεγονός ότι τα RNN που έχουν εκπαιδευτεί με BPTT δυσκολεύονται να μάθουν μακροπρόθεσμες εξαρτήσεις (π.χ. εξαρτήσεις μεταξύ βημάτων που απέχουν πολύ) εξαιτίας αυτού που ονομάζεται πρόβλημα διαφυγής (Vanishing/Exploding Gradient Problem). Το πρόβλημα αυτό προτάθηκε από τους Bengio et al. (1994). Υπάρχουν ορισμένοι μηχανισμοί για την αντιμετώπιση αυτών των προβλημάτων και ορισμένοι τύποι RNN (όπως τα LSTM) που σχεδιάστηκαν ειδικά για να τα ξεπεράσουν τέτοιου είδους προβλήματα.

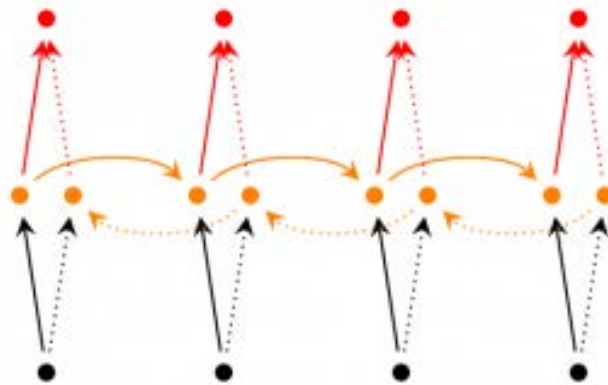
### **2.6.2 Επεκτάσεις των RNNs**

Με τα χρόνια οι ερευνητές έχουν αναπτύξει πιο εξελιγμένους τύπους RNNs για να αντιμετωπίσουν μερικές από τις ελλείψεις του μοντέλου RNN vanilla. Στην συνέχεια θα παρουσιάσουμε λεπτομερέστερα κάποιους από αυτούς τους τύπους RNNs.

#### Αμφίδρομα RNNs (Bidirectional RNNs)

Τα αμφίδρομα RNNs βασίζονται στην ιδέα ότι η έξοδος στο χρόνο  $t$  μπορεί όχι μόνο να εξαρτάται από τα προηγούμενα στοιχεία της ακολουθίας αλλά και από τα μελλοντικά στοιχεία. Για παράδειγμα, για να προβλέψουμε μια λέξη που λείπει σε μια ακολουθία, μπορεί να θέλουμε να δούμε τόσο στο αριστερό όσο και στο δεξί άκρο της “περιοχής” της λέξης που μας ενδιαφέρει. Τα αμφίδρομα RNNs είναι αρκετά απλά. Είναι μόνο δύο RNN στοιβαγμένα το ένα πάνω στο άλλο. Η έξοδος υπολογίζεται έπειτα με βάση την κρυφή κατάσταση και των δύο RNNs.

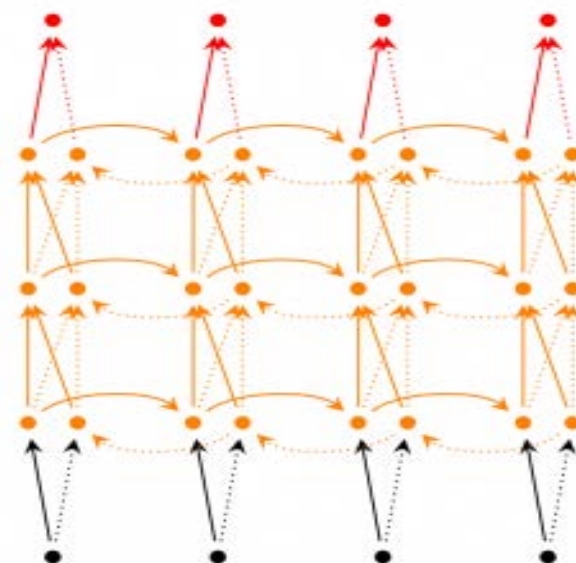
Στην Εικόνα 3.13 παρουσιάζεται η σχηματική αναπαράσταση ενός αμφίδρομου RNN.



**Εικόνα 2.13: Σχηματική αναπαράσταση ενός αμφίδρομου RNN.**  
 (Εικόνα από: <http://poodar.me/Bidirectional-RNNs/>)

Βαθιά (αμφίδρομα) RNNs (Deep (Bidirectional) RNNs)

Τα βαθιά (αμφίδρομα) RNNs είναι παρόμοια με τα αμφίδρομα RNNs, μόνο που τώρα έχουμε πολλαπλά κρυμμένα στρώματα για κάθε χρονικό βήμα. Στην πράξη αυτό μας δίνει μια υψηλότερη ικανότητα εκμάθησης (αλλά χρειαζόμαστε επίσης πολλά στοιχεία κατάρτισης).



**Εικόνα 2.14: Σχηματική αναπαράσταση ενός Deep Bidirectional RNN.**  
 (Εικόνα από: <http://poodar.me/Bidirectional-RNNs/>)





### LSTM networks

Τα LSTM Networks είναι αρκετά δημοφιλή στις μέρες μας και θα ασχοληθούμε εκτενέστερα με αυτή την κατηγορία νευρωνικών δικτύων στην συνέχεια αυτής της εργασίας. Τα LSTMs δεν έχουν ουσιαστική διαφορά στην αρχιτεκτονική τους από τα RNN, αλλά χρησιμοποιούν μια διαφορετική συνάρτηση για να υπολογίσουν την κρυφή κατάσταση. Η μνήμη στα LSTM ονομάζεται κελιά (cells) και μπορούμε να τα θεωρήσουμε ως μαύρα πλαίσια που λαμβάνουν ως είσοδο την προηγούμενη κατάσταση  $h_{t-1}$  και την τρέχουσα είσοδο  $x_t$ . Εσωτερικά αυτά τα κύτταρα αποφασίζουν τι πρέπει να διατηρούν (και τι να διαγράψουν) από τη μνήμη. Στη συνέχεια, συνδυάζουν την προηγούμενη κατάσταση, την τρέχουσα μνήμη και την είσοδο. Αποδεικνύεται ότι αυτός ο τύπος νευρωνικών δικτύων είναι πολύ αποδοτικός στη λήψη μακροπρόθεσμων εξαρτήσεων κάτι στο οποίο υστερούν τα RNNs.

### 2.6.3 Εφαρμογές των RNNs

Τα RNNs χρησιμοποιούνται με μεγάλη επιτυχία σε πολλά NLP προβλήματα. Σε αυτό το σημείο θα αναφέρουμε μερικά παραδείγματα εφαρμογών των RNNs σε NLP.

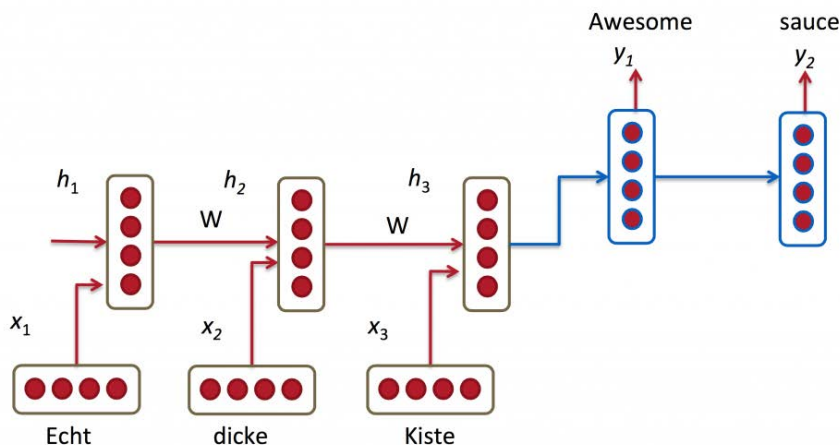
- Μοντελοποίηση γλωσσών και δημιουργία κειμένου (Language Modeling and Generating Text)

Λαμβάνοντας υπ όψιν μια ακολουθία λέξεων, μπορούμε να προβλέψουμε την πιθανότητα κάθε λέξης με δεδομένες τις προηγούμενες λέξεις. Τα Μοντέλα Γλωσσών μας επιτρέπουν να μετρήσουμε πόσο πιθανή είναι η εμφάνιση μιας πρότασης, κάτι το οποίο είναι πολύ σημαντικό στην συνεισφορά στην Μηχανική Μετάφραση (επειδή οι προτάσεις υψηλής πιθανότητας είναι συνήθως σωστές). Μια παρενέργεια της δυνατότητας να προβλέψουμε την επόμενη λέξη είναι ότι έχουμε ένα γενικό μοντέλο, το οποίο μας επιτρέπει να δημιουργούμε νέο κείμενο με δειγματοληψία από τις πιθανότητες εξόδου. Έτσι ανάλογα με τα δεδομένα εκπαίδευσης που διαθέτουμε, μπορούμε να δημιουργήσουμε όλους τους συνδυασμούς. Στο μοντέλο γλωσσών η είσοδός μας είναι συνήθως μια ακολουθία λέξεων (κωδικοποιημένη σαν ένα input vector) και η έξοδος μας είναι η ακολουθία των προβλεπόμενων λέξεων. Κατά την εκπαίδευση του δικτύου θέτουμε  $o_t = x_{t+1}$  αφού θέλουμε η έξοδος στο βήμα  $t$  να είναι η πραγματική επόμενη λέξη.

- Machine Translation

Το machine translation ή αλλιώς αποκωδικοποιητής γλώσσας είναι παρόμοιο με την προηγούμενη εφαρμογή μόνου που εδώ έχουμε μια ακολουθία από λέξεις (πρόταση) σε μια γλώσσα πχ. Γερμανικά και θέλουμε σαν έξοδο να έχουμε μια ακολουθία από λέξεις σε μια διαφορετική γλώσσα, για παράδειγμα Αγγλικά.

Μια βασική διαφορά είναι ότι η παραγωγή μας αρχίζει μόνο αφού έχουμε δει την πλήρη είσοδο, διότι η πρώτη λέξη των μεταφρασμένων προτάσεών μας μπορεί να απαιτεί πληροφορίες που συλλέγονται από την πλήρη ακολουθία εισόδου.



**Εικόνα 2.15: RNN για Machine Translation.**

(Εικόνα από: <http://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf>)

- Αναγνώριση Φωνής

Δεδομένης της ακολουθίας εισόδου ακουστικών σημάτων από ένα ηχητικό κύμα, μπορούμε να προβλέψουμε μια ακολουθία φωνητικών τμημάτων μαζί με τις πιθανότητες τους.

- Δημιουργία περιγραφής εικόνων

Μαζί με τα CNNs, τα RNN χρησιμοποιήθηκαν ως μέρος ενός μοντέλου για τη δημιουργία περιγραφών για μη επισημασμένες εικόνες. Είναι εκπληκτικό το πόσο καλά φαίνεται αυτό να λειτουργεί. Το συνδυασμένο μοντέλο ευθυγραμμίζει ακόμα και τις δημιουργούμενες λέξεις με χαρακτηριστικά που βρίσκονται στις εικόνες.

### 2.6.4 Backpropagation Through Time (BPTT) και Vanishing Gradients

Σε προηγούμενη παράγραφο αναλύσαμε τα RNNs και αναφέραμε τον όρο BPTT. Σε αυτή την παράγραφο θα αναλύσουμε αυτή την διαδικασία και θα προσπαθήσουμε να καταλάβουμε το πρόβλημα του Vanishing Gradient το οποίο ουσιαστικά συνέβαλλε στην ανάπτυξη των LSTM Networks.

Ανακεφαλαιώνοντας θα παραθέσουμε τις βασικές εξισώσεις που διέπουν τα RNNs. Η μοναδική διαφορά μόνο, για λόγους βιβλιογραφίας, θα είναι πως την έξοδο του νευρωνικού δικτύου θα την συμβολίζουμε με  $\hat{y}$ , αντί για  $o$ .

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = \text{softmax}(Vs_t)$$

Επίσης, ορίσαμε την απώλεια ή σφάλμα μας ως:



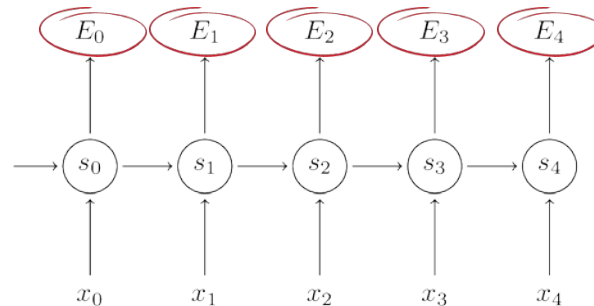


$$E_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t$$

$$E(y, \hat{y}) = \sum_t E_t(y_t, \hat{y}_t)$$

$$= - \sum_t y_t \log \hat{y}_t$$

Στην παραπάνω εξίσωση  $y_t$  είναι η σωστή τιμή του αντικειμένου που θέλουμε να προβλέψουμε την χρονική στιγμή  $t$  και  $\hat{y}_t$  είναι η προβλεπόμενη τιμή για την χρονική στιγμή  $t$ . Συνήθως θεωρούμε όλη την αλληλουχία σαν ένα παράδειγμα εκμάθησης και έτσι το ολικό σφάλμα είναι το άθροισμα των σφαλμάτων για κάθε χρονική στιγμή.



Ο στόχος μας είναι να υπολογίσουμε τις διαβαθμίσεις του σφάλματος σε σχέση με τις παραμέτρους μας  $U$ ,  $V$  και  $W$  και στη συνέχεια να μάθουμε τις “καλές” παραμέτρους χρησιμοποιώντας το Stochastic Gradient Descent. Ακριβώς όπως συνοψίζουμε τα σφάλματα, συνοψίζουμε επίσης τις κλίσεις (gradients) σε κάθε χρονική στιγμή για ένα παράδειγμα εκπαίδευσης:

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}$$

Για να υπολογίσουμε αυτές τις κλίσεις χρησιμοποιούμε τον κανόνα της αλυσίδας. Αυτό δεν είναι παρά ο αλγόριθμος που χρησιμοποιούμε στο backpropagation όταν εφαρμόζεται προς τα πίσω ξεκινώντας από το σφάλμα. Έστω για την χρονική στιγμή  $t=3$  θα έχουμε:

$$\frac{\partial E_3}{\partial V} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial V}$$

$$= \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial z_3} \frac{\partial z_3}{\partial V}$$

$$= (\hat{y}_3 - y_3) \otimes s_3$$

όπου  $z_3 = V s_3$  και  $\otimes$  το εξωτερικό γινόμενο των δυο διανυσμάτων. Το βασικό συμπέρασμα ανάμεσα σε αυτές τις εξισώσεις είναι πως η κλίση για την χρονική στιγμή 3 (όμοια και για όλες τις υπόλοιπες χρονικές στιγμές) εξαρτάται μόνο από  $\hat{y}_3$ ,  $y_3$  και  $s_3$  για το διάνυσμα  $V$ .

Τώρα για το διάνυσμα  $U$  γράφοντας τον κανόνα της αλυσίδας όπως και πριν έχουμε τα παρακάτω:

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial W}$$

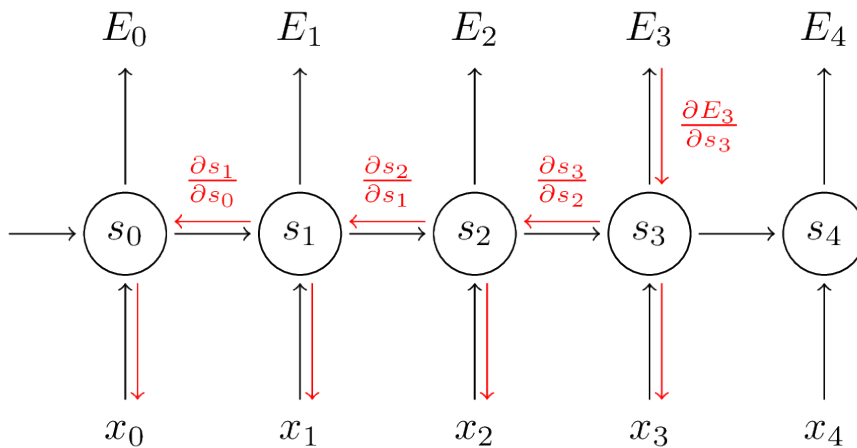
Σε αυτή την περίπτωση παρατηρούμε ότι  $s_3 = \tanh(U x_t + W s_2)$ , δηλαδή το  $s_3$  εξαρτάται από το  $s_2$ . Γι αυτό τον λόγο δεν μπορούμε να συμπεριφερθούμε στη μεταβλητή  $s_2$  σαν μια



σταθερά. Θα πρέπει να εφαρμόσουμε τον κανόνα της αλυσίδας ξανά. Το αποτέλεσμα φαίνεται στην παρακάτω εξίσωση:

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$

Στην πραγματικότητα αθροίζουμε την συνεισφορά του κάθε χρονικού βήματος στην κλίση (gradient). Αυτό συμβαίνει επειδή το  $W$  χρησιμοποιείται σε κάθε χρονική στιγμή μέχρι την χρονική στιγμή 3 πρέπει να κάνουμε backpropagation σε όλα τα gradients στο δίκτυο από την χρονική στιγμή 3, προς τα πίσω, μέχρι και την χρονική στιγμή 0.



Μια σημαντική παρατήρηση είναι ότι αυτό είναι ακριβώς το ίδιο με τον τυπικό αλγόριθμο backpropagation που χρησιμοποιούμε στα Feed Forward Νευρωνικά Δίκτυα. Η βασική διαφορά είναι ότι αθροίζουμε τις κλίσεις για το  $W$  σε κάθε βήμα. Σε ένα παραδοσιακό NN δεν μοιραζόμαστε παραμέτρους μεταξύ των επιπέδων, οπότε δεν χρειάζεται να αθροίσουμε τίποτα.

### 2.6.5 Το πρόβλημα του *Vanishing Gradient*

Σε προηγούμενη παράγραφο αναφέραμε πως τα RNNs έχουν πρόβλημα στο να “μάθουν” μακροσκελείς αλληλουχίες, για παράδειγμα αλληλεπιδράσεις μεταξύ λέξεων που βρίσκονται αρκετά βήματα μακριά. Για να καταλάβουμε το γιατί ας ξαναδούμε την κλίση που υπολογίσαμε πιο πριν:

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$

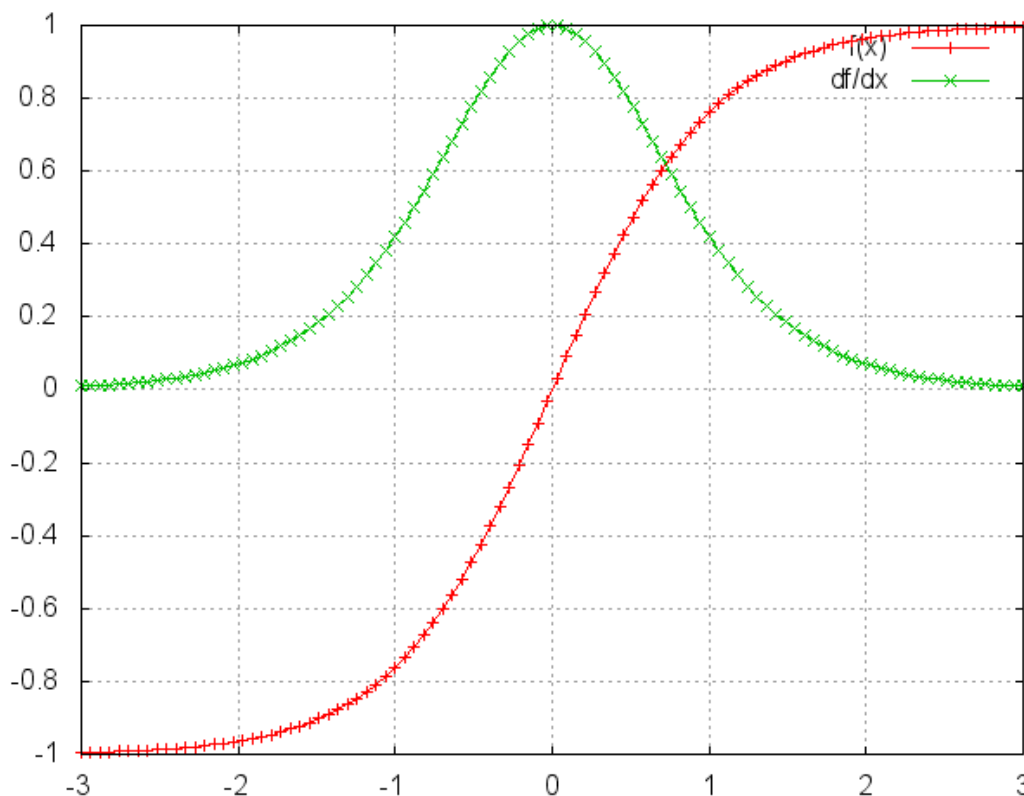
Αν παρατηρήσουμε καλύτερα θα δούμε ότι η μερική παράγωγος του  $s_3$  εμπεριέχει ένα κανόνα αλυσίδας από μόνη της.

$$\frac{\partial s_3}{\partial s_k} \implies \frac{\partial s_3}{\partial s_1} = \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial s_1}$$

Σαν αποτέλεσμα, λόγω του ότι παίρνουμε την παράγωγο μιας διανυσματικής συνάρτησης, θα έχουμε έναν Ιακωβιανό πίνακα όπου τα στοιχεία του θα είναι όλες οι μερικές παράγωγοι του διανύσματος. Φαίνεται πως η νόρμα του παραπάνω Ιακωβιανού πίνακα έχει ένα άνω όριο



στο 1 (paper source: *On the difficulty of training recurrent neural networks*, Razvan Pascanu, Tomas Mikolov, Yoshua Bengio).



**Εικόνα 2.16: Γραφικές παραστάσεις της συνάρτησης tanh και της παραγώγου.**  
 (Εικόνα από: <http://nn.readthedocs.org/en/rtd/transfer/>)

Μπορούμε να δούμε ότι οι συνάρτηση tanh καθώς και η σιγμοειδής συνάρτηση προσεγγίζουν μια επίπεδη γραμμή. Όταν συμβεί αυτό, λέμε ότι οι αντίστοιχοι νευρώνες έχουν πλέον κορεστεί. Έχουν μηδενική κλίση δηλαδή και οδηγούν τις υπόλοιπες κλίσεις, σε προηγούμενα στρώματα, προς το 0. Αυτό έχει σαν αποτέλεσμα μετά από μερικά βήματα οι τιμές των κλίσεων να συρρικνώνονται και τελικά να εξαφανίζονται τελείως μετά από μερικά χρονικά βήματα. Έτσι η συνεισφορά των κλίσεων από πολύ μακρινές χρονικές στιγμές γίνεται μηδενική με αποτέλεσμα το νευρωνικό μας δίκτυο να μην μαθαίνει σωστά. Αυτό το πρόβλημα δεν παρουσιάζεται αποκλειστικά στα RNNs. Συμβαίνουν επίσης σε νευρωνικά δίκτυα τύπου Feed Forward, απλά επειδή τα RNNs τείνουν να είναι πολύ βαθιά, αυτό το πρόβλημα είναι αρκετά κοινό.

Μια από τις μεθόδους ελαχιστοποίησης του προβλήματος αυτού είναι η σωστή αρχικοποίηση του πίνακα  $W$ . Άλλη μια λύση στο πρόβλημα του Vanishing Gradient είναι η χρήση της συνάρτησης ReLU, αντί για τις tanh και σιγμοειδή συνάρτηση, σαν συνάρτηση ενεργοποίησης του νευρωνικού δικτύου. Τέλος άλλη μια λύση, ίσως και η προτιμότερη, είναι η χρήση των LSTM (Long Short Term Memory) ή Gated Recurrent Unit (GRUs) νευρωνικών που θα παρουσιάσουμε στην επόμενη παράγραφο.



## 2.7 LSTM Neural Networks

Όπως έχουμε αναφέρει πολλές φορές στην προηγούμενη ενότητα τα LSTM (Long Short Term Memory) νευρωνικά δίκτυα αποτελούν μια ειδική κατηγορία των νευρωνικών δικτύων RNN και είναι ικανά να μπορούν να μάθουν μακροσκελείς ακολουθίες. Αυτού του τύπου νευρωνικά δίκτυα προτάθηκαν αρχικά από τους Hochreiter και Schmidhuber το 1997 και χρησιμοποιήθηκαν και εξελίχθηκαν από πολλούς ανθρώπους στην συνέχεια. Φαίνονται να δουλεύουν πολύ καλά και χρησιμοποιούνται σε μια πληθώρα προβλημάτων στις μέρες μας.

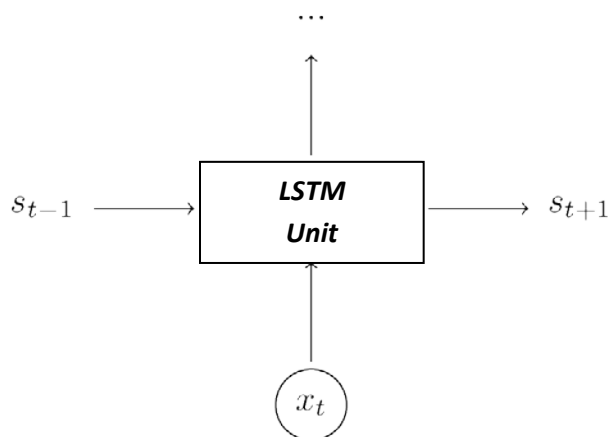
Τα LSTMs είναι ειδικά σχεδιασμένα για να αποφευχθεί το πρόβλημα της εκμάθησης μακροσκελών ακολουθιών και η μεταξύ τους εξάρτηση. Πιο συγκεκριμένα το να “θυμούνται” πληροφορία που βρίσκεται σε πολύ μακρινές χρονικές στιγμές είναι ο λόγος που σχεδιάστηκαν και όχι κάτι για το οποίο πασχίζουν να διεκπεραιώσουν όπως τα RNNs.

Για να καταλάβουμε ακριβώς πως δουλεύουν τα LSTMs αρχικά θα δούμε πως υπολογίζεται η κρυφή κατάσταση  $s_t$  σε τέτοιου είδους δίκτυα.

$$\begin{aligned}
 i &= \sigma(x_t U^i + s_{t-1} W^i) \\
 f &= \sigma(x_t U^f + s_{t-1} W^f) \\
 o &= \sigma(x_t U^o + s_{t-1} W^o) \\
 g &= \tanh(x_t U^g + s_{t-1} W^g) \\
 c_t &= c_{t-1} \circ f + g \circ i \\
 s_t &= \tanh(c_t) \circ o \longrightarrow s_t = \tanh(U x_t + W s_{t-1})
 \end{aligned}$$

Σύμφωνα με τις προηγούμενες εξισώσεις βλέπουμε πως οι εισοδοί για την κρυφή κατάσταση είναι το  $x_t$  (η είσοδος την χρονική στιγμή  $t$ ) και το  $s_{t-1}$  (η κρυφή κατάσταση την προηγούμενη χρονική στιγμή  $t-1$ ). Η έξοδος έχει και αυτή μια κρυφή κατάσταση  $s_t$ ...

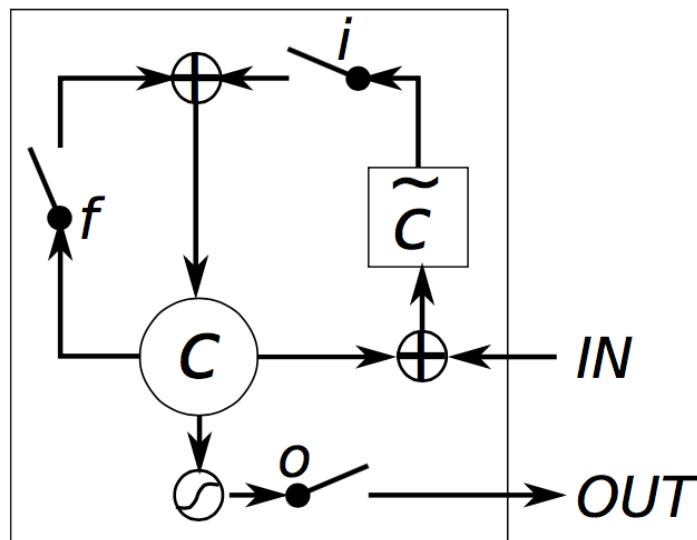
Το συμπέρασμα είναι ότι μπορούμε να αντιλαμβανόμαστε τις δομικές μονάδες ενός LSTM σαν ξεχωριστά κουτιά (black boxes όπως αναφέρονται στην βιβλιογραφία) όπου δοθέντος μιας εισόδου και της προηγούμενης κρυφής κατάστασης υπολογίζουν την επόμενη κρυφή κατάσταση.





Συνοψίζοντας θα δώσουμε μια περιγραφή των όρων και μεταβλητών που χρησιμοποιήθηκαν στις προηγούμενες εξισώσεις για καλύτερη κατανόηση:

- Τα  $i$ ,  $f$ ,  $o$  ονομάζονται input, forget και output πύλες. Αν παρατηρήσουμε καλύτερα θα δούμε ότι περιγράφονται ακριβώς από τις ίδιες εξισώσεις αλλά έχουν διαφορετικούς πίνακες μεταβλητών. Ονομάζονται πύλες γιατί η σιγμοειδής συνάρτηση συμπίεζει τις τιμές των διανυσμάτων αυτών μεταξύ του 0 και του 1 και πολλαπλασιάζοντας τα με κάποιο άλλο διάνυσμα μπορούμε να ελέγξουμε πόσο θα επηρεάσει αυτό το διάνυσμα την έξοδο, ή αλλιώς τι ποσοστό από αυτό το διάνυσμα θα περάσει. Η πύλη εισόδου καθορίζει πόσο από τη νέα υπολογισμένη κατάσταση για την τρέχουσα είσοδο θέλουμε να περάσει. Η πύλη forget πόση από την προηγούμενη κατάσταση θέλουμε να περάσει και τέλος, η πύλη εξόδου ορίζει το ποσοστό της εσωτερικής κατάστασης θέλετε να περάσει στο εξωτερικό δίκτυο (υψηλότερα επίπεδα και το επόμενο βήμα). Όλες οι πύλες έχουν τις ίδιες διαστάσεις, δηλαδή έχουν ίδιο μέγεθος κρυφών καταστάσεων.
- $g$  είναι η υποψήφια κρυφή κατάσταση η οποία υπολογίζεται με βάση την είσοδο για την τρέχουσα χρονική στιγμή και την κρυφή κατάσταση της προηγούμενης χρονικής στιγμής.
- $c_t$  είναι η εσωτερική μνήμη της μονάδας (unit). Είναι ένας συνδυασμός της εσωτερικής μνήμης της προηγούμενης χρονικής στιγμής πολλαπλασιασμένη με την πύλη forget και της νέας κρυφής κατάστασης  $g$  πολλαπλασιασμένη με την πύλη εισόδου. Για να το προσεγγίσουμε διαισθητικά είναι ένας συνδυασμός του πως θέλουμε να “ενώσουμε” την μνήμη της χρονικής στιγμής  $t-1$  με την είσοδο της χρονικής στιγμής  $t$ .
- Δεδομένου του ότι έχουμε υπολογίσει την εσωτερική μνήμη  $c_t$ , μπορούμε να υπολογίσουμε την κρυφή κατάσταση εξόδου  $st$  πολλαπλασιάζοντας την εσωτερική μνήμη με την πύλη εξόδου. Πρέπει να σημειωθεί πως μπορεί να μην είναι όλα τα στοιχεία της εσωτερικής μνήμης σχετικά με την κρυφή κατάσταση που χρησιμοποιείται από άλλες μονάδες του δικτύου.



**Εικόνα 2.17: Πύλες στο LSTM Δίκτυο**

(Εικόνα από: <http://nn.readthedocs.org/en/rtd/transfer/>) Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." (2014)

### 2.7.1 GRU Μια Διαφορετική προσέγγιση στα LSTMs

Η ιδέα πίσω από τα GRU είναι παρόμοια με αυτή των LSTMs και διέπεται από τους παρακάτω κανόνες:

$$z = \sigma(x_t U^z + s_{t-1} W^z)$$

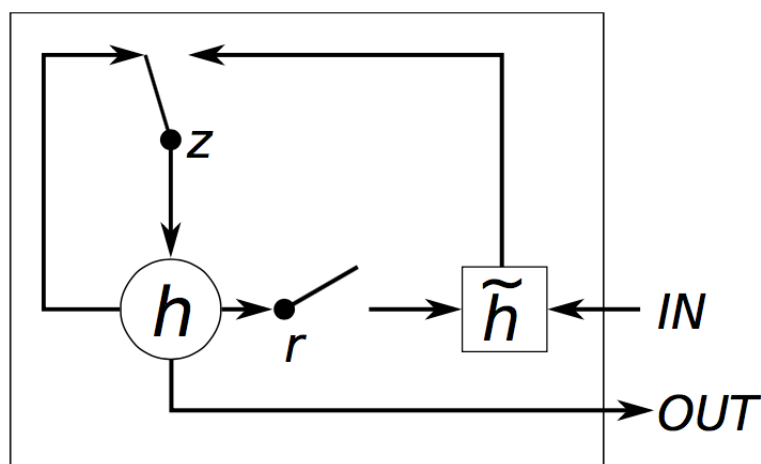
$$r = \sigma(x_t U^r + s_{t-1} W^r)$$

$$h = \tanh(x_t U^h + (s_{t-1} \circ r) W^h)$$

$$s_t = (1 - z) \circ h + z \circ s_{t-1}$$

Ένα GRU έχει δύο πύλες, μία πύλη reset  $r$  και μία πύλη update  $z$ . Η πύλη reset αποφασίζει το πώς θα συνδυάζει την νέα είσοδο με την προηγούμενη μνήμη και η πύλη update καθορίζει το πόσο από την προηγούμενη μνήμη θα χρησιμοποιηθεί και θα διατηρηθεί. Ένα κρατήσουμε στην reset μόνο τις τιμές 1 και στην update μόνο τις τιμές 0 τότε έχουμε ένα δίκτυο RNN. Η βασική ιδέα στο να χρησιμοποιήσουμε ένα GRU για την εκμάθηση μακροσκελών ακολουθιών είναι ίδια με αυτή των LSTMs με μερικές διαφοροποιήσεις.

- Τα GRUs έχουν δύο πύλες ενώ τα LSTMs έχουν τρεις πύλες.
- Τα GRUs δεν έχουν εσωτερική μνήμη  $c_t$  η οποία είναι διαφορετική από την κρυφή κατάσταση. Ακόμα δεν έχουν την πύλη εξόδου (output) που υπάρχει στα LSTMs.
- Η πύλη εισόδου (input) και η πύλη forget έχουν αντικατασταθεί από την πύλη update  $z$  και η πύλη reset  $r$  εφαρμόζεται κατευθείαν στην προηγούμενη κρυφή κατάσταση.



**Εικόνα 2.17: Πύλες στο GRU Δίκτυο** (Εικόνα από: Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." (2014))

Τώρα που έχουμε δει τα δύο μοντέλα για να εξαλείψουμε το πρόβλημα της μείωσης της κλίσης, είναι λογικό να αναρωτηθεί κανείς: Ποιο θα χρησιμοποιήσω; Τα GRUs είναι αρκετά καινούργια (2014) και τα μειονεκτήματά τους δεν έχουν εξερευνηθεί πλήρως. Σύμφωνα με εμπειρικές αξιολογήσεις στα άρθρα Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling και An Empirical Exploration of Recurrent Network Architectures, δεν υπάρχει σαφής νικητής. Σε πολλές εργασίες και οι δύο αρχιτεκτονικές παράγουν συγκρίσιμες επιδόσεις και η ρύθμιση παραμέτρων και των μεταβλητών όπως το μέγεθος των layers είναι

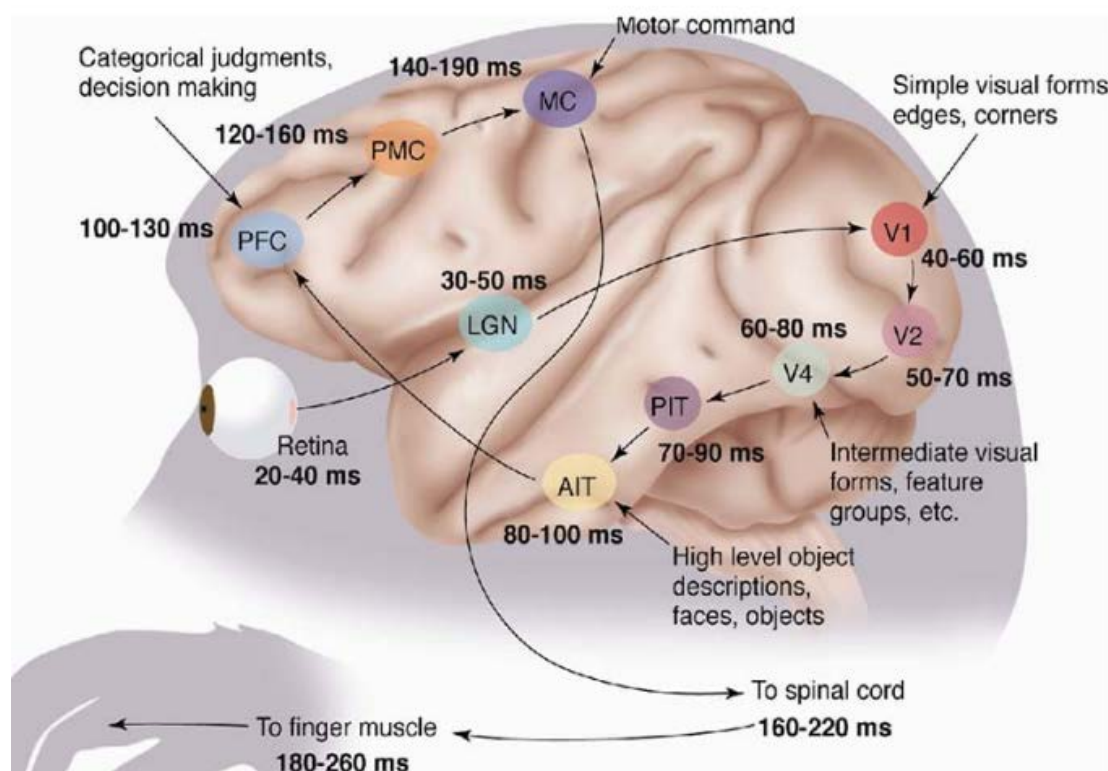


πιθανότητα πιο σημαντική από την επιλογή της ιδανικής αρχιτεκτονικής. Οι μονάδες GRU έχουν λιγότερες παραμέτρους (U και W είναι μικρότερες) και έτσι μπορούν να εκπαιδευτούν λίγο πιο γρήγορα ή γενικότερα χρειάζονται λιγότερα δεδομένα. Από την άλλη πλευρά, στην περίπτωση που έχουμε αρκετά δεδομένα, η μεγαλύτερη εκφραστική δύναμη των LSTM μπορεί να οδηγήσει σε καλύτερα αποτελέσματα.

## 2.8 Deep Learning Neural Networks

### 2.8.1 Ορισμός και Ιδιότητες των Βαθιών Αρχιτεκτονικών

Μια βασική διαφορά των ρηχών (shallow) αρχιτεκτονικών νευρωνικών δικτύων από τις βαθιές (deep) αρχιτεκτονικές είναι η χρησιμοποίηση πολλών επιπέδων στην δομή του δικτύου και όχι μόνο δυο ή τριών, που, βεβαίως σύμφωνα με τα θεωρητικά αποτελέσματα επαρκούν για την αντιστοίχιση των δεδομένων εισόδου στις επιθυμητές αποκρίσεις του δικτύου στην έξοδο. Τα δίκτυα αυτά προσομοιώνουν καλύτερα τις λειτουργίες της ανθρώπινης οπτικής αντίληψης, όπως φαίνεται στην Εικόνα 2.18, όπου αποτυπώνονται τα πολλαπλά επίπεδα λειτουργίας και οι ενδιάμεσες αναπαραστάσεις που δημιουργούνται, από την σύλληψη της εικόνας στον αμφιβληστροειδή χιτώνα μέχρι την ανθρώπινη αντίδραση στο ερέθισμα μέσω των μυών των χεριών.



**Εικόνα 2.18: Πολλαπλά επίπεδα λειτουργίας και ενδιάμεσες αναπαραστάσεις του συστήματος οπτικής αντίληψης** (Εικόνα από: <https://emmarobinson01.com/2016/05/05/clever-robots-understanding-artificial-intelligence-by-looking-at-our-own-brains/>).

Στην Εικόνα αυτή, κάθε μονάδα μετατρέπει την αναπαράσταση εισόδου της σε μια υψηλότερου επιπέδου. Τα χαρακτηριστικά υψηλού επιπέδου είναι πιο γενικά και πιο αμετάβλητα, ενώ τα χαρακτηριστικά χαμηλού επιπέδου βοηθούν στην κατηγοριοποίηση των εισόδων.





Τα τεχνητά συστήματα που προσομοιώνουν τέτοιες λειτουργίες έχουν ως στόχο να μαθαίνουν να δημιουργούν τις απαραίτητες ενδιάμεσες αναπαραστάσεις και να παράγουν με επιτυχία την τελική τους εκτίμηση.

Τα συστήματα αυτά μπορεί να βασίζονται σε μια από τις επόμενες τεχνικές:

- Επιβλεπόμενη Μάθηση

Η αρχικοποίηση των παραμέτρων είναι τυχαία. Η εκπαίδευση είναι επιβλεπόμενη, συνήθως τύπου στοχαστικής κλίσης καθόδου, με χρησιμοποίηση του αλγορίθμου *back propagation* για υπολογισμό των κλίσεων, ενώ χρησιμοποιείται στα περισσότερα πρακτικά συστήματα για την αναγνώριση ομιλίας και εικόνας.

- Μη επιβλεπόμενη μάθηση με επιβλεπόμενο ταξινομητή στην έξοδο

Η μέθοδος εκπαιδεύει κάθε επίπεδο χωρίς επίβλεψη, το ένα μετά το άλλο, ενώ τελικά εκπαιδεύει έναν επιβλεπόμενο ταξινομητή στην κορυφή, διατηρώντας τα άλλα στρώματα σταθερά. Η μέθοδος έχει καλή απόδοση όταν είναι διαθέσιμα πολύ λίγα κατηγοριοποιημένα δείγματα.

- Μη επιβλεπόμενη μάθηση με επιβλεπόμενο εξομαλυντή

Η μέθοδος εκπαιδεύει κάθε επίπεδο χωρίς επίβλεψη, το ένα μετά το άλλο, προσθέτει ένα στρώμα ταξινομητή, και επανεκπαιδεύει όλο το σύστημα με επίβλεψη. Έχει καλή απόδοση όταν το σύνολο των ετικετών είναι φτωχό (π.χ. για ανίχνευση πεζών).

Ένα ερώτημα το οποίο μπορεί να σκεφτεί κανείς είναι ότι αφού μπορούμε και προσεγγίζουμε μια συνάρτηση όσο κοντά θέλουμε με χρήση ρηχών αρχιτεκτονικών γιατί να μπει κάποιος στην διαδικασία να χρησιμοποιήσει τις βαθιές αρχιτεκτονικές νευρωνικών δικτύων (*Deep Learning*). Είναι γνωστό ότι τα νευρωνικά δίκτυα με 2 επίπεδα επιτυγχάνουν καθολική προσέγγιση των συναρτήσεων εισόδου εξόδου.

$$y = \sum_{i=1}^p a_i K(X, X^i)$$

$$F(W^1, F(W^0, X))$$

Οι μηχανές βαθιάς Μάθησης

$$y = F(W^k, F(W^{k-1}, F(\dots F(W^1, F(W^0, X))\dots)))$$

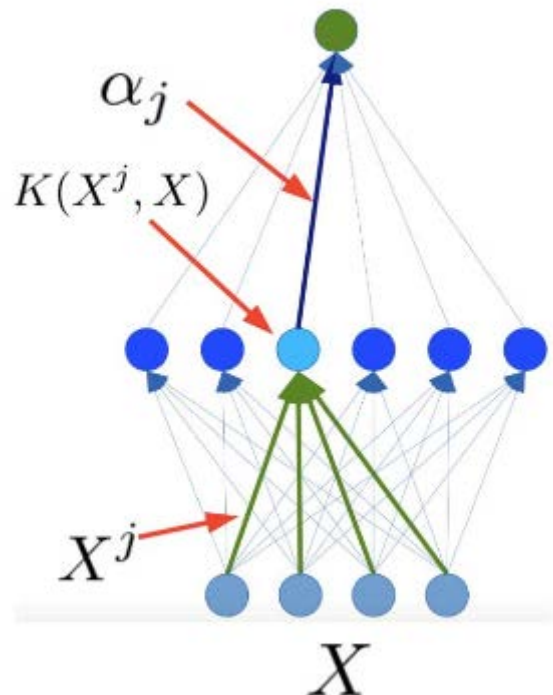
είναι πιο αποδοτικές για αναπαράσταση ορισμένων κατηγοριών συναρτήσεων, ιδιαίτερα εκείνων που συμμετέχουν στην οπτική αναγνώριση, καθόσον μπορούν να αντιπροσωπεύσουν περισσότερο πολύπλοκες συναρτήσεις με λιγότερο "υλικό" (hardware), σε πολλές εργασίες τεχνητής νοημοσύνης (υπολογιστική όραση, υπολογιστική ακοή, επεξεργασία φυσικής γλώσσας).

Τα μοντέλα 2 επιπέδων δεν είναι βαθιά (ακόμη και αν εκπαιδεύσουμε το πρώτο επίπεδο), επειδή δεν υπάρχει ιεραρχία χαρακτηριστικών. Ομοίως τα νευρωνικά δίκτυα με ένα κρυφό επίπεδο δεν είναι βαθιά, όπως και οι μηχανές διανυσμάτων υποστήριξης και οι μέθοδοι πυρήνα που εξετάζουμε στο επόμενο Κεφάλαιο. Τα ίδια ισχύουν και για τα δέντρα ταξινόμησης, όπου επίσης δεν υπάρχει ιεραρχία χαρακτηριστικών. Όλες οι αποφάσεις λαμβάνονται στο χώρο εισόδου. Μια τέτοια αρχιτεκτονική φαίνεται στο Σχήμα 2.19.





$$G(X, a) = \sum_j a_j K(X^j, X)$$



Εικόνα 2.19: Μια μηχανή διανυσμάτων υποστήριξης με βάση συναρτήσεις πυρήνα.

Η βαθιά μάθηση εμπεριέχει μη κυρτές συναρτήσεις σφάλματος. Η κυρτή μάθηση δεν εξαρτάται από την σειρά με την οποία παρουσιάζονται τα δείγματα (εξαρτάται μόνο από τις ασυμπτωτικές συχνότητες του δείγματος). Η ανθρώπινη μάθηση δεν είναι έτσι... μαθαίνουμε απλές έννοιες πριν μάθουμε πολύπλοκες. Η σειρά με την οποία μαθαίνουμε πράγματα έχει σημασία.

### 2.8.2 Restricted Boltzman Machines (RBMs)

Ας υποθέσουμε πως έχουμε ένα training set από δυαδικά διανύσματα τα οποία για λόγους επεξηγηματικούς και χάριν ευκολίας θα πούμε ότι είναι δυαδικές εικόνες. Το σύνολο εκπαίδευσης μπορεί να μοντελοποιηθεί χρησιμοποιώντας ένα δίκτυο δύο στρωμάτων που λέγεται “Restricted Boltzman Machine” (Smolensky, 1986; Freund and Haussler, 1992; Hinton, 2002). Τα δυαδικά εικονοστοιχεία συνδέονται με στοχαστικούς, δυαδικούς εξαγωγείς χαρακτηριστικών (feature detectors) χρησιμοποιώντας συνδέσεις ίδιου βάρους. Τα εικονοστοιχεία αντιστοιχούν στα “ορατά” στοιχεία του RBM γιατί μπορούμε να παρατηρήσουμε τα χαρακτηριστικά τους. Αντίθετα οι εξαγωγείς χαρακτηριστικών αντιστοιχούν στα “κρυφά” στοιχεία του RBM. Η ένωση  $(v, h)$  των ορατών στοιχείων και των κρυφών έχει μια ενέργεια (Hopfield, 1982) η οποία δίνεται από την σχέση:

$$E(v, h) = - \sum_{i \in \text{visible}} a_i u_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} h_j u_i w_{ij}$$

όπου  $h_j, u_i$  είναι οι δυαδικές καταστάσεις του ορατού στοιχείου  $i$  και του  $j$  κρυφού,  $a_i, b_j$  είναι οι αποκλίσεις (bias) τους και  $w_{ij}$  είναι το βάρος στην μεταξύ τους σύνδεση. Το δίκτυο θέτει μια πιθανότητα σε κάθε δυνατό ζευγάρι ορατού και κρυφού διανύσματος μέσω της παρακάτω συνάρτησης ενέργειας.



$$p(v, h) = \frac{1}{Z} e^{-E(v,h)}$$

όπου η Z (partition function) δίνεται αθροίζοντας όλα τα δυνατά ζεύγη των ορατών και κρυμμένων διανυσμάτων:

$$Z = \sum_{v,h} e^{-E(v,h)}$$

Η πιθανότητα η οποία αποδίδεται σε ένα διάνυσμα ορατού στοιχείου v δίνεται από την παρακάτω σχέση:

$$p(v) = \frac{1}{Z} \sum_h e^{-E(v,h)}$$

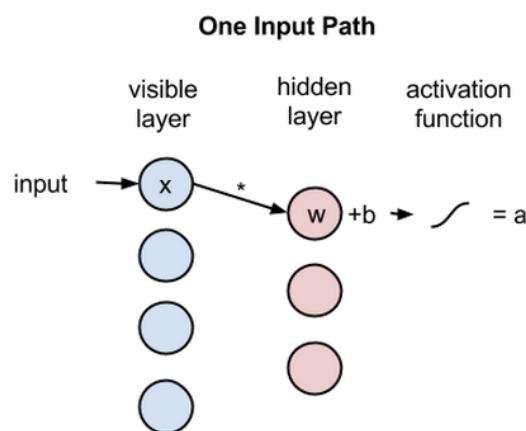
Η πιθανότητα η οποία αποδίδεται από το δίκτυο σε μια εικόνα εκπαίδευσης μπορεί να αυξηθεί, προσαρμόζοντας τα βάρη και τις αποκλίσεις με σκοπό να μειωθεί η ενέργεια αυτής της εικόνας και ταυτόχρονα να αυξηθεί η ενέργεια άλλων εικόνων, ιδίως εκείνων που έχουν χαμηλές ενέργειες και ως εκ τούτου έχουν μια μεγάλη συνεισφορά στην συνάρτηση Z (partition function).

$$\frac{d \log p(v)}{dw_{ij}} = \langle h_j u_i \rangle_{data} - \langle h_j u_i \rangle_{model}$$

Με το learning rate να δίνεται από

$$\Delta w_{ij} = (\langle h_j u_i \rangle_{data} - \langle h_j u_i \rangle_{model})$$

Σε μια πιο απλή προσέγγιση στην συνέχεια παρουσιάζεται η λειτουργία ενός RBM σχηματικά.



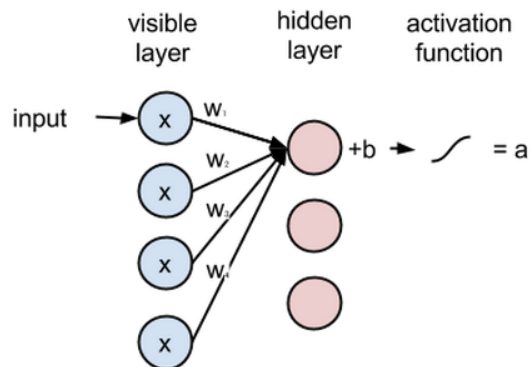
Κάθε κύκλος στο παραπάνω γράφημα αντιπροσωπεύει μια μονάδα νευρώνα, ονομάζεται κόμβος, και οι κόμβοι είναι απλά όπου οι υπολογισμοί λαμβάνουν χώρα. Οι κόμβοι συνδέονται μεταξύ τους σε όλη τα στρώματα, αλλά δεν υπάρχουν δύο κόμβοι του ίδιου στρώματος οι οποίοι συνδέονται μεταξύ τους.

Κάθε κόμβος του ορατού στρώματος λαμβάνει ένα χαρακτηριστικό χαμηλού επιπέδου από ένα στοιχείο στο σύνολο δεδομένων που πρέπει να μάθει. Για παράδειγμα, για ένα σύνολο δεδομένων από ασπρόμαυρες εικόνες, κάθε κόμβος του ορατού στρώματος ή στρώματος εισόδου θα λάβει μια τιμή για κάθε pixel σε μία εικόνα. (Εικόνες MNIST έχουν 784 pixels, με αποτέλεσμα τα νευρωνικά δίκτυα θα πρέπει να έχουν 784 κόμβους εισόδου στο ορατό στρώμα.).



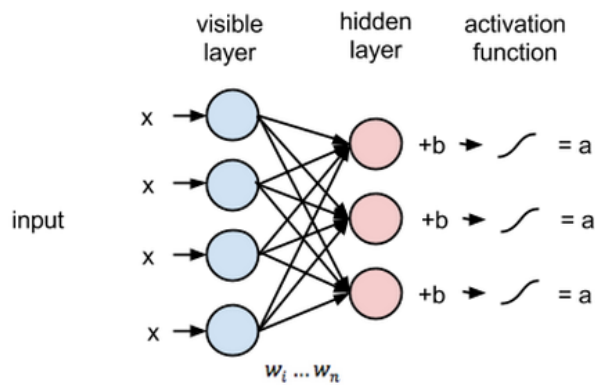
Έτσι για πολλαπλές εισόδους έχουμε την παρακάτω κατάσταση:

### Weighted Inputs Combine @Hidden Node



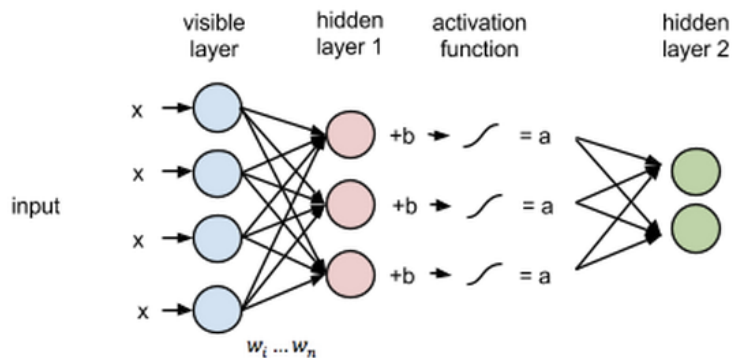
Όλοι οι κόμβοι εισόδου συνδέονται με όλους τους κόμβους του πρώτου κρυφού στρώματος υπολογίζοντας τα βάρη για κάθε κατάσταση.

### Multiple Inputs



Κάθε κόμβος του κρυφού στρώματος λαμβάνει τις τέσσερις εισόδους πολλαπλασιάζονται με τον αντίστοιχο βάρους τους. Το άθροισμα τους και πάλι προστίθεται στην απόκλιση (που αναγκάζει τουλάχιστον κάποιες ενεργοποιήσεις να συμβούν), και το αποτέλεσμα διέρχεται μέσω του αλγορίθμου ενεργοποίησης έτσι ώστε να παράγει μία έξοδο για κάθε κρυφό κόμβο.

### Multiple Hidden Layers



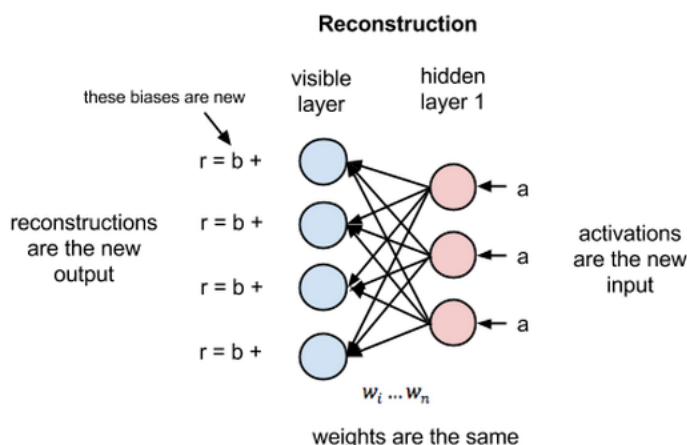


Αν αυτά τα δύο στρώματα ήταν μέρος ενός deep neural network, οι έξοδοι του πρώτου κρυφού στρώματος θα είναι οι εισροές του δεύτερου κρυφού στρώματος και από εκεί μέσω όλων των hidden layers που θα έχει το μοντέλο μας θα οδηγούμασταν στο στρώμα εξόδου του δικτύου μας.

### Ανακατασκευή Δεδομένων στα RBMs

Σε αυτό το κομμάτι θα εξηγήσουμε την διαδικασία κατά την οποία τα RBMs μαθαίνουν να ανακατασκευάζουν δεδομένα χωρίς εποπτεία (unsupervised learning).

Στη φάση ανασυγκρότησης, τα αποτελέσματα του πρώτου κρυφού στρώματος γίνονται οι εισόδοι για το backward pass. Πολλαπλασιάζονται με τα ίδια βάρη, ένα για κάθε σύνδεση μεταξύ των κόμβων, ακριβώς με την ίδια διαδικασία που γίνεται και στο forward pass. Το άθροισμα αυτών των γινομένων προστίθεται με τις αποκλίσεις (bias) του ορατού στρώματος κάθε ορατού κόμβου, και η έξοδος των πράξεων αυτών αποτελεί την ανακατασκευή, δηλαδή μια προσέγγιση της αρχικής εισόδου. Αυτό μπορεί να παρασταθεί με τον ακόλουθο διάγραμμα:



Λόγω του ότι τα βάρη των RBMs αρχικοποιούνται τυχαία η διαφορά μεταξύ των ανακατασκευασμένων δεδομένων και των αρχικών δεδομένων εισόδου είναι συνήθως μεγάλη. Αυτό το σφάλμα μέσω του συνεχόμενου back propagation μειώνεται μέχρι να επιτευχθεί η ελάχιστη δυνατή διαφορά.

Στην εμπρόσθια τροφοδότηση τα RBMs χρησιμοποιούν τις εισόδους στο ορατό στρώμα για να κάνουν προβλέψεις στην έξοδο του κρυφού στρώματος βάσει της δεσμευμένης πιθανότητας  $p(a|x; w)$ . Στην αντίθετη περίπτωση, δηλαδή στο προς τα πίσω πέρασμα από το κρυφό προς το ορατό στρώμα, η μηχανή μας προσπαθεί να υπολογίσει τις πιθανότητες των εισόδων δοθέντων των outputs που δίνει το κρυφό στρώμα εξόδου.

Πολλά RBMs μαζί δημιουργούν μια δομή η οποία ονομάζεται CRBM (Continuous Restricted Boltzmann Machine). Η μορφή αυτή έχει την δυνατότητα να δέχεται συνεχόμενες εισόδους μέσω ενός διαφορετικού τύπου CD Sampling. Αυτό επιτρέπει στο CRBM να χειριστεί δεδομένα όπως pixel από εικόνες τα οποία κανονικοποιούνται σε δεκαδικά ψηφία από το μηδέν μέχρι το ένα.

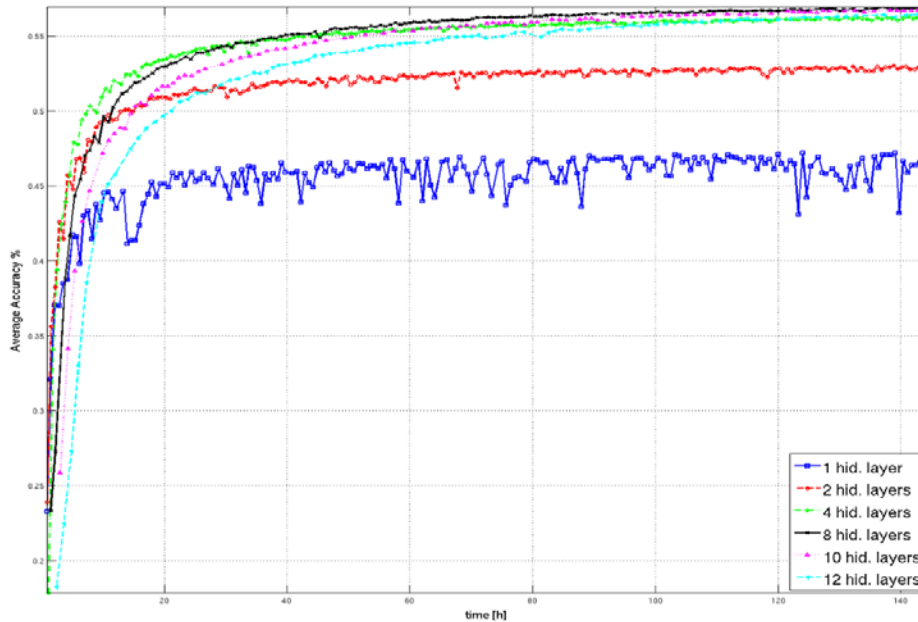
Θα πρέπει να σημειωθεί πως κάθε δίκτυο deep learning χρειάζεται τέσσερα δομικά βασικά στοιχεία.

- Την είσοδο
- Τους Συντελεστές



- Το bias (απόκλιση)
- Και τον μετασχηματισμό – συνάρτηση ενεργοποίησης

Στην συνέχεια παρουσιάζεται ένα γράφημα το οποίο παρουσιάζει το σφάλμα σε συνάρτηση με τα layers τα οποία χρησιμοποιούνται σε ένα deep learning δίκτυο:



Παρατηρούμε πως για μεγαλύτερο αριθμό κρυφών στρωμάτων μπορούμε να πετύχουμε μεγαλύτερη ακρίβεια στα αποτελέσματά μας. Βέβαια από κάποιο σημείο και μετά μπορούμε να δούμε πως οι καμπύλες επικαλύπτονται πράγμα που μας οδηγεί στο συμπέρασμα πως υπάρχει ένας κορεσμός στον αριθμό των κρυφών στρωμάτων που μπορούμε να χρησιμοποιήσουμε έτσι ώστε αυτά να είναι αποδοτικά.



## Κεφάλαιο 3 – Τεχνικές Πρόβλεψης

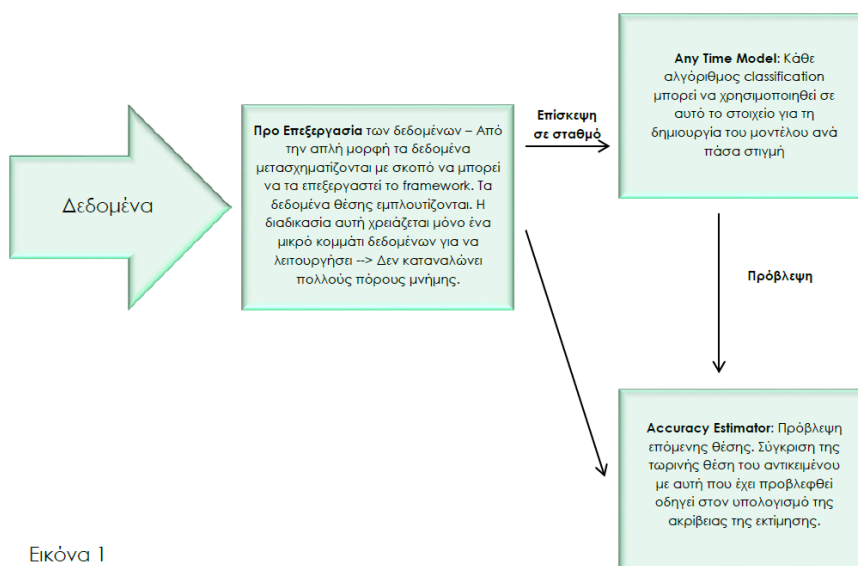
### 3.1 – Τεχνικές Πρόβλεψης θέσης χωρίς χρήση Νευρωνικών Δικτύων

Για την κίνηση και πρόβλεψη της τροχιάς κινούμενων αντικειμένων έχουν γίνει πολλές μελέτες από την επιστημονική κοινότητα (τόσο την ελληνική όσο και την παγκόσμια). Υπάρχουν πολλές δημοσιευμένες εργασίες που αναπτύσσουν τεχνικές για την πρόβλεψη της μελλοντικής θέσης αντικειμένων και σε αυτό το κεφάλαιο θα παρουσιαστούν συνοπτικά κάποιες από αυτές καθώς και οι μέθοδοι οι οποίες χρησιμοποιήθηκαν.

Συνήθως ένα κινούμενο αντικείμενο είναι εξοπλισμένο με μια συσκευή μετάδοσης η οποία περιοδικά σηματοδοτεί τη θέση του στον αντίστοιχο ασύρματο δέκτη που συλλέγει την πληροφορία. Η μέθοδος η οποία θα παρουσιαστεί είναι ανεξάρτητη από την τοπολογία της κίνησης του αντικειμένου και έχει πολύ καλή συμπεριφορά με την αύξηση του αριθμού των κινούμενων σωμάτων, πράγμα που συνήθως αποτελεί πρόβλημα για αντίστοιχους αλγόριθμους πρόβλεψης θέσης. Ο αλγόριθμος δημιουργεί ένα index παρόμοιο με αυτό του FP – Tree (Frequent Pattern Tree). Το δέντρο εμπλουτίζεται κάθε φορά που ο αλγόριθμος εντοπίζει ένα trajectory το οποίο εμφανίζεται συχνά. Με αυτό τον τρόπο το index περιέχει όλες τις συχνές διαδρομές για γρήγορη αναζήτηση. Από τη στιγμή που έχουμε βρει όλα τα συχνά trajectories μπορούμε να τα χρησιμοποιήσουμε για την πρόβλεψη της θέσης ενός κινούμενου αντικειμένου. Από κει και πέρα, αφού έχουν δημιουργηθεί οι αντίστοιχοι κανόνες συσχέτισης, για κάθε κινούμενο αντικείμενο συγκρίνεται το μέχρι τώρα trajectory του με τους κανόνες συσχέτισης που έχουμε εξάγει από τα frequent patterns που έχουν αποθηκευτεί στο index. (Morzy\_2007)

Οι περισσότερες εργασίες που αφορούν την πρόβλεψη της μελλοντικής θέσης χρησιμοποιούν μοντέλα classification χρησιμοποιώντας παλαιότερα δεδομένα για το training του μοντέλου. Η μέθοδος που έχει χρησιμοποιηθεί εδώ [Joao Bartolo Gomes, Clifton Phua, Shonali Krishnaswamy - Mobile Data Mining for Next Place Prediction] μεταφέρει όλη την διαδικασία εκμάθησης στην κινητή συσκευή του χρήστη (κινητό, GPS κτλ) χωρίς να χρειαστεί να κοινοποιήσει τα προσωπικά δεδομένα. Το framework το οποίο χρησιμοποιείται σε αυτό το πείραμα μοντελοποιεί το πρόβλημα σαν ένα πρόβλημα classification το οποίο εκτελείται πχ στο κινητό τηλέφωνο του χρήστη.

Στην Εικόνα 1 παρουσιάζεται μια σχεδιαστική απεικόνιση του framework και των λειτουργιών του.



Εικόνα 1



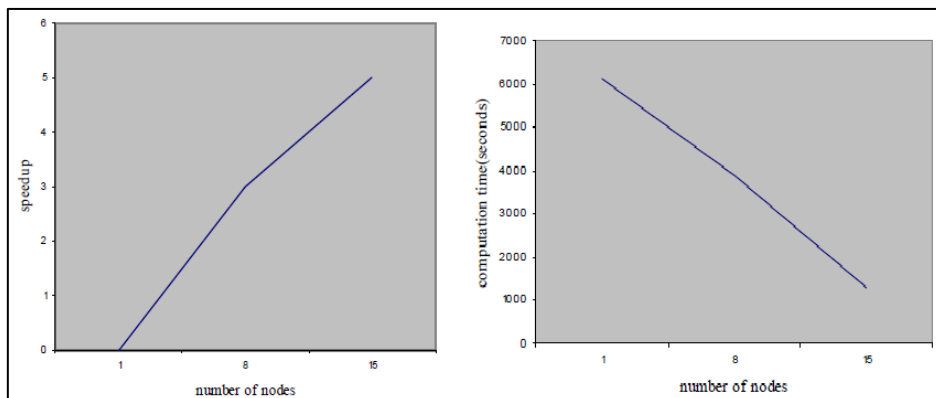
Τα αποτελέσματα έδειξαν πως σε κάποιους χρήστες των οποίων οι τροχιές είναι αρκετά ακανόνιστες είναι πολύ δύσκολο να έχουμε κάποια πρόβλεψη. Από την άλλη πλευρά, σε ποιο ομαλές κινήσεις είναι δυνατόν να επιτευχθεί ακρίβεια περισσότερη από 80%.

(*Trajectory Clustering for Motion Prediction\_Sung-IROS12*) Σε αυτή την εργασία προτείνεται ένας πρωτότυπος αλγόριθμος για την εξαγωγή μοτίβων κίνησης από διάφορες τροχιές δεδομένων και γίνεται σύγκριση των αποτελεσμάτων αυτών σε σχέση με το τον κλασσικό αλγόριθμο clustering k-means. Ο αλγόριθμος αποτελείται από 4 βήματα.

- **Line Simplification:** Σε αυτό το κομμάτι γίνεται η τμηματοποίηση του trajectory κόβοντας το στα κρίσιμα του σημεία. Κρίσιμα σημεία ορίζονται ως τα σημεία εκείνα στα οποία αλλάζει η συμπεριφορά της κίνησης του αντικειμένου. Στα πειράματα που έγιναν στα πλαίσια αυτού του paper χρησιμοποιήθηκε μια εναλλακτική μορφή του αλγόριθμου Douglas – Peucker. Έτσι δημιουργούνται κάποια sub trajectories.
- **k – Lines Projection:** Σε αυτό το βήμα ο αλγόριθμος ανάγει το πρόβλημα σε μονοδιάστατο υπολογίζοντας τις αντίστοιχες προβολές των sub trajectories στην μια διάσταση.
- **Interval Clustering:** Στο τρίτο βήμα έχοντας υπολογίσει τις προβολές από το Βήμα 2 προχωρούμε στην ομαδοποίηση (clustering) των sub trajectories.
- **Calculation of Representatives:** Από τη στιγμή που στο προηγούμενο βήμα έχουν εξαχτεί τα clusters των sub trajectories αντιστοιχείται σε κάθε ένα από αυτά μια αντιπροσωπευτική γραμμή (πορεία).

Από τα πειράματα φαίνεται ότι ο αλγόριθμος παράγει πολύ καλά αποτελέσματα σε αντίθεση με τον αλγόριθμο ομαδοποίησης k-means ο οποίος φαίνεται να χάνει τμήματα τροχιών οι οποίες δεν είναι και τόσο συχνές.

(*Mobility Prediction of Mobile Users in Mobile Environment Using Knowledge Grid*) Στην συγκεκριμένη εργασία προσεγγίζεται το πρόβλημα της πρόβλεψης της μελλοντικής θέσης με έναν διαφορετικό τρόπο. Η κίνηση ενός κινούμενου αντικειμένου ονομάζεται UAP – User Actual Path και έχει τη μορφή  $\langle I_1, I_2, \dots, I_n \rangle$  όπου n ο αριθμός των τοποθεσιών που έχουν ακολουθηθεί από τον χρήστη (κινούμενο αντικείμενο) και όπου  $k = 1, 2, 3, \dots$  η κοστή τοποθεσία του κινούμενου αντικειμένου. Οι κινήσεις των αντικειμένων μας (δεδομένα από τα κινητά τηλέφωνα των χρηστών) αποθηκεύονται σε κόμβους οι οποίοι επικοινωνούν ασύρματα με τους πομπούς. Η περιοχή κάλυψης ολόκληρου του δικτύου χωρίζεται σε ένα data grid. Ο αλγόριθμος εξόρυξης γνώσης εκτελείται στα δεδομένα του πλέγματος με σκοπό να εξαγάγει τα trajectories τα οποία έχουν χρησιμοποιηθεί συχνά από τους χρήστες (UMP – User Mobility Pattern) με σκοπό να παράγει του αντίστοιχούς κανόνες κίνησης – συσχέτισης. Ο αλγόριθμος ο οποίος χρησιμοποιείται είναι ο KMPM. Σημαντικό χαρακτηριστικό του συγκεκριμένου καταναμημένου αλγόριθμου είναι η πολύ καλή απόκριση με την αύξηση των κόμβων (speedup). Στην Εικόνα 2 παρουσιάζεται αυτό με την μορφή διαγραμμάτων.







(*An Approach to Predicting the Location of Moving Objects During On-Road Navigation*) Σε προηγούμενα papers δεν λαμβάνεται υπ όψιν η μορφολογία της τοποθεσίας στην οποία διεξάγεται η κίνηση καθώς και άλλα τυχόν αντικείμενα. Σε αυτό το paper παρουσιάζονται οι μηχανισμοί για την κατάλληλη εφαρμογή αλγορίθμων πρόβλεψης κίνησης με σκοπό να έχουμε πλήρη προσαρμογή με την παρουσία κινούμενων εμποδίων στην τροχιά του υπό εξέταση κινούμενου αντικειμένου. Όσον αφορά την μορφολογία του εδάφους στο οποίο διεξάγεται η κίνηση (off road ή on road διαδρομές) χρησιμοποιούνται και διαφορετικοί αλγόριθμοι πρόβλεψης κίνησης. Αυτό βέβαια προϋποθέτει την γνώση και ύπαρξη a priori χαρτών που περιέχουν χαρακτηριστικά του εδάφους η real time δεδομένα τα οποία θα δημιουργούν την εκάστοτε μορφολογία βάσει γνώσης η οποία θα προέρχεται από δίκτυα αισθητήρων. Ακόμη υπάρχουν αποθηκευμένα δεδομένα, χωρισμένα σε κλάσεις, για τα κινούμενα αντικείμενα – εμπόδια τα οποία συναρτώνται στον δρόμο όπως οχήματα, ζώα, πεζοί, συντρίμια κτλ. Βάσει αυτών των πληροφοριών τα αποτελέσματα των αλγορίθμων πρόβλεψης απεικονίζονται σε ένα time based πλέγμα.

Άλλη μια μέθοδος εξόρυξης γνώσης για την πρόβλεψη θέσης σε κινούμενα αντικείμενα παρουσιάζεται στην εργασία *A data mining approach for location prediction in mobile environments*. Στην εργασία αυτή, προτείνεται ένας νέος αλγόριθμος για την πρόβλεψη της επόμενης θέσης μεταξύ των κυψελών κίνησης ενός κινητού χρήστη σε ένα προσωπικό δίκτυο Επικοινωνιακών Συστημάτων (PCS). Αποτελείται από 3 στάδια:

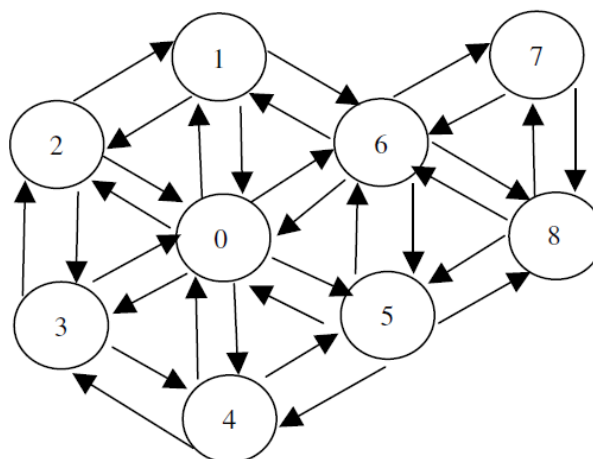
- Αρχικά εξορύσσονται από τις τροχιές παλιότερων χρηστών τα χαρακτηριστικά της κίνησης του προς εξέταση αντικειμένου.
- Δημιουργούνται οι κανόνες συσχέτισης από τα προηγούμενα πρότυπα.
- Τέλος με την χρήση αυτών των κανόνων επιτυγχάνεται η πρόβλεψη της θέσης των αντικειμένων.

Η προβλεπόμενη κίνηση μπορεί τότε να χρησιμοποιηθεί για την αύξηση της αποτελεσματικότητας της PCSs.

Οι πρώτες δύο (2) φάσεις του αλγόριθμου τρέχουν offline. Ωστόσο, η τελευταία φάση είναι πραγματοποιείται online. Αυτό σημαίνει ότι κάθε φορά που ένας χρήστης προτίθεται να κάνει μια κίνηση, μια αίτηση αποστέλλεται στο σύστημα και η πρόβλεψη γίνεται χρησιμοποιώντας τον αλγόριθμο πρόβλεψης. Η κίνηση ενός αντικειμένου από, το κελί του δικτύου στο οποίο βρίσκεται σε ένα άλλο, καταγράφεται σε μια βάση δεδομένων η οποία ονομάζεται home location register (**HLR**). Επιπρόσθετα κάθε σταθμός – βάση κρατάει μια βάση δεδομένων με τα στοιχεία των χρηστών που βρίσκονται σε αυτό το κελί. Αυτή η βάση δεδομένων ονομάζεται visitor location register (VLR). Με αυτά τα δεδομένα είναι εφικτό να εξαγάγουμε την κίνηση που έχει κάνει το αντικείμενο κοιτώντας τα logs που έχουν γίνει στο HLR. Σε αυτή την εργασία τα trajectories είναι της μορφής

$T = \langle (id1, t1), (id2, t2), \dots, (idk, tk) \rangle$  όπου το id1 δηλώνει το ID του κελιού στο οποίο εισέρχεται το αντικείμενο τη χρονική στιγμή t1. Στις εγγραφές δύο διαδοχικοί αριθμοί του ID αντιστοιχούν πρέπει να αντιστοιχούν σε γειτονικά κελιά. Τα κελιά του δικτύου απεικονίζονται με την μορφή κατευθυνόμενων γράφων όπως φαίνεται στην Εικόνα 4.





Εικόνα 3.1 – Μορφή κελιών Δικτύου

Εάν ένα αντικείμενο μείνει παραπάνω χρόνο από ένα καθορισμένα threshold σε ένα κελί με ID πχ id1 τότε δημιουργείται ένα νέο δημιουργείται ένα νέο subsequence trajectory το οποίο καταγράφεται και αυτό. Αυτά τα trajectories τα οποία έχουν δημιουργηθεί ονομάζονται user actual paths (UAPs). Θεωρούμε τα UAPs ως πολύτιμη πηγή πληροφοριών, διότι η κινητικότητα των χρηστών περιέχει τόσο τακτικά όσο και τυχαία μοτίβα. Ως εκ τούτου, με τη χρήση των UAPs, μπορεί να είμαστε σε θέση να εξάγουμε τα συχνά μοτίβα και να τα χρησιμοποιήσουμε στην πρόβλεψη. Τα συχνά χρησιμοποιούμενα μοτίβα ονομάζονται user mobility patterns (UMP). Η εξόρυξη γνώσης από τα UMPs μας επιτρέπει να εξάγουμε τους κανόνες συσχέτισης.

Στην εργασία WhereNext: a Location Predictor on Trajectory Pattern Mining η πρόβλεψη της κίνηση ενός αντικειμένου βασίζεται στην υπόθεση ότι οι άνθρωποι τείνουν να ακολουθούν το πλήθος. Δηλαδή ακολουθούν μονοπάτια τα οποία χρησιμοποιούνται συχνά και από πολύ κόσμο. Για παράδειγμα, οι άνθρωποι πηγαίνουν στη δουλειά τους κάθε μέρα με παρόμοιες διαδρομές και τα μέσα μαζικής μεταφοράς διασχίζουν παρόμοιες διαδρομές σε διαφορετικές χρονικές στιγμές. Η μέθοδος που χρησιμοποιήθηκε σε αυτή την εργασία χωρίζεται σε τέσσερα (4) μέρη:

- **Data Selection** (Επιλογή Δεδομένων): Αρχικά επιλέγονται τα trajectories αυτά τα οποία αντιστοιχούν στην χωρική περιοχή και χρονική περίοδο που μας ενδιαφέρει να εξετάσουμε.
- **Local Models Extraction**: Από τα trajectories που έχουν επιλεγεί από το προηγούμενο στάδιο εκτελείται ο αλγόριθμος με σκοπό να βρεθούν τα πιο συχνά μοτίβα κίνησης τα οποία ονομάζονται trajectory patterns.
- **T – Pattern Tree Building**: Όλα τα μοτίβα από το προηγούμενο βήμα συνδέονται με σκοπό τη δημιουργία ενός δέντρου (T – pattern Tree). Οι κόμβοι του δέντρου είναι περιοχές που συχνά έχουν επισκεφθεί και οι πλευρές αντιπροσωπεύουν τα ταξίδια μεταξύ τους.
- **Prediction**: Το δένδρο το οποίο δημιουργήθηκε χρησιμοποιείται για την πρόβλεψη της μελλοντικής τοποθεσίας του κινούμενου αντικειμένου.

Για την διαδικασία της πρόβλεψης όπως αναφέραμε και προηγουμένως χρησιμοποιείται το δένδρο το οποίο έχει δημιουργηθεί. Η βασική ιδέα είναι ότι δοθέντος ενός αρχικού trajectory γίνεται η προσπάθεια να ταυτοποιηθεί αυτό με ένα ήδη υπάρχον από το δένδρο το οποίο του



μοιάζει όσο το δυνατόν περισσότερο. Η μέθοδος ενέχει πολλούς κινδύνους αλλά εάν ξεπεραστούν αποτελεί έναν καλό τρόπο για την πρόβλεψη της μελλοντικής θέσης.

Ένας αλγόριθμος ο οποίος είναι ήδη γνωστός για τη χρήση του σε δεδομένα χρονοσειρών είναι ο ARIMA (Auto – Regression Integrated Moving Average). Η προσπάθεια που έχει γίνει είναι να επεκταθεί ο αλγόριθμος και για χωρικά δεδομένα. Η μέθοδος ARIMA για τις χρονοσειρές θεωρείται από τις καλύτερες γραμμικές μεθόδους. Σε αυτή την εργασία γίνεται η προσπάθεια να δοθεί και η χωρική διάσταση στον αλγόριθμο για network constrained trajectory data. Η αρχική διαδικασία είναι η μετατροπή των αρχικών trajectories σε χρονοσειρές τις οποίες μπορεί να διαχειριστεί ο αλγόριθμος. Τα δεδομένα μετασχηματίζονται από την μορφή εισαγωγής  $\langle xt, yt, ti \rangle$  (μορφή αρχικού trajectory) σε  $\langle s1, t1 \rangle, \langle s2, t2 \rangle, \dots, \langle sn, tn \rangle$  (trajectory speed time series) και σε  $\langle d1, t1 \rangle, \langle d2, t2 \rangle, \dots, \langle dn, tn \rangle$  (other time series model about distances). Επόμενο βήμα είναι η επιλογή του κατάλληλου μοντέλου που υπολογίζει τις παραμέτρους ARIMA(p,d,q). Οι προβλέψεις που προκύπτουν είναι αρκετά κοντά στην πραγματικότητα παρόλ' αυτά υπάρχουν και προβλέψεις οι οποίες είναι μακριά από την πραγματικότητα λόγω δύο (2) κυρίως λόγων:

- Παρουσία outliers
- Αλλαγή της συμπεριφοράς του οχήματος

Τέλος στην εργασία 'Aircraft Trajectory Prediction Made Easy with Predictive Analytics' περιγράφεται μια τεχνική για την πρόβλεψη τροχιάς αεροσκαφών. Εδώ βλέπουμε μια διαφορετική προσέγγιση αφού πολύ σημαντικό ρόλο στην μελέτη της εργασίας παίζει τόσο το υψόμετρο αλλά και οι καιρικές συνθήκες που επικρατούν. Η προσέγγιση αυτή "βλέπει" τον εναέριο χώρο σαν ένα 3D grid με κάθε σημείο του grid να αντικατοπτρίζει μια περιοχή παρατήρησης των καιρικών φαινομένων. Σαν αποτέλεσμα δημιουργούνται υποθετικοί κύβοι γύρω από αυτά τα σημεία έτσι ώστε να καλυφθεί ολόκληρος ο εναέριος χώρος από αυτά (Οι καιρικές συνθήκες μένουν αμετάβλητες μέσα σε κάθε κύβο κατά την διάρκεια μιας χρονικής περιόδου). Έτσι συνδυάζοντας τους κύβους με τα trajectories δημιουργείται ένα 4D grid (longitude, altitude, latitude, time). Η εκπαίδευση του μοντέλου γίνεται βάσει ιστορικών γεγονότων ,και λαμβάνοντας υπ' όψιν τις καιρικές συνθήκες, χρησιμοποιείται το μοντέλο Hidden Markov Model με σκοπό να βρεθεί το πιο πιθανό trajectory που θα ακολουθήσει το αεροσκάφος. Η προσέγγιση στην εύρεση της πιο πιθανής διαδρομής σε αυτό το paper διαφοροποιείται από το υπόλοιπα όσον αφορά τα παρακάτω σημεία:

- ✓ Χρησιμοποιείται μια πιθανολογική προσέγγιση λαμβάνοντας υπ' όψιν και τις διάφορες αβεβαιότητες με σκοπό να εξασφαλισθεί μεγαλύτερη ακρίβεια στο μοντέλο.
- ✓ Τα trajectories θεωρούνται ως μια σειρά από κύβους 4ων διαστάσεων, πραγματοποιείται δηλαδή η ενοποίηση των trajectories με τις καιρικές μεταβλητές.
- ✓ Πραγματοποιείται η τεχνική time series clustering της οποίας τα αποτελέσματα τροφοδοτούν τον αλγόριθμο Viterbi\* ο οποίος αποτελεί τον βέλτιστο τρόπο με σκοπό την πρόβλεψη.
- ✓ Η χρήση real time δεδομένων με σκοπό τον έλεγχο της αποδοτικότητας της προσέγγισης.



**Εικόνα 3.2:** δεξιά- 2D αναπαράσταση του grid καιρικών συνθηκών , μέση- 3D αναπαράσταση των aligned trajectories\* (κίτρινο) και των raw trajectories\*, αριστερά - κύβοι από χωροχρονικά δεδομένα

\*raw trajectory: Μια πεπερασμένη ακολουθία από timestamps που έχει εξαχθεί δειγματοληπτικά από το αρχικό trajectory.

\*aligned trajectory: Ένα σύνολο σημείων αναφοράς που προέρχονται από τον μετασχηματισμό των raw trajectories.

\* **Αλγόριθμος Viterbi**

Ο αλγόριθμος Viterbi είναι ένας δυναμικός αλγόριθμος προγραμματισμού για την εύρεση της πιο πιθανής ακολουθίας διαφόρων κρυμμένων καταστάσεων (Viterbi Path) δεδομένων των πραγματικών καταστάσεων. Ο αλγόριθμος έχει ευρύ πεδίο εφαρμογών όπως είναι η αναγνώριση ομιλίας, η υπολογιστική γλωσσολογία και οι τηλεπικοινωνίες ενώ συναντώνται εφαρμογές και στη βιοπληροφορική καθώς το πρόβλημα που λύνει βρίσκει εφαρμογή σε πολλούς κλάδους της τεχνολογίας.

Έστω μια ακολουθία παρατηρούμενων συμβόλων  $y_1, y_2, \dots, y_T$   $T$  στοιχείων με καταστάσεις στο πεπερασμένο σύνολο  $S$ ,  $\pi_i$  η αρχική πιθανότητα της κατάστασης  $i$  και  $a_{i,j}$  οι πιθανότητες μετάβασης από την κατάσταση  $i$  στην κατάσταση  $j$ .

Η πιθανότερη ακολουθία  $x_1, x_2, \dots, x_T$  από την οποία προέκυψαν οι παρατηρήσεις προκύπτει από τις σχέσεις:

$$V_{1,k} = P(y_1 | k) \cdot \pi_k$$

$$V_{t,k} = \max_{x \in S} (P(y_t | k) \cdot a_{x,k} \cdot V_{t-1,x})$$

όπου  $V_{t,k}$  είναι η πιθανότητα της πιο πιθανής ακολουθίας καταστάσεων για τις πρώτες  $t$  παρατηρήσεις που καταλήγει στην κατάσταση  $k$ .

Όπως φαίνεται, οι σχέσεις αυτές δεν δίνουν την πιθανότερη ακολουθία καταστάσεων. Για να γίνει αυτό ο αλγόριθμος έχει ένα δεύτερο βήμα που υπολογίζει το μονοπάτι προς τα πίσω:



$$x_t = \arg \max_{x \in S} (V_{t,x})$$

$$x_{t-1} = Ptr(x_t, t)$$

Δηλαδή, πρώτα βρίσκει ο αλγόριθμος την τελική κατάσταση ως αυτή με τη μέγιστη τελική πιθανότητα και στη συνέχεια ανακατά τις καταστάσεις προς τα πίσω από την κατάσταση που χρησιμοποιήθηκε στη σχέση υπολογισμού των πιθανοτήτων  $V_{t,k}$ .

### 3.3 – Τεχνικές Πρόβλεψης Θέσης με Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα και πιο συγκεκριμένα τα RNNs, τα LSTMs καθώς και οι παραλλαγές τους, στις οποίες αναφερθήκαμε εκτενώς στο προηγούμενο κεφάλαιο μπορούν να χρησιμοποιηθούν για την πρόβλεψη μελλοντικών καταστάσεων αντικειμένων (όπως αυτοκινήτων, αεροσκαφών, ακόμα και πεζών ανθρώπων), την εύρεση ακολουθίας λέξεων μέσα σε μια πρόταση και πολλά άλλα.

Το πλεονέκτημα της χρήσης των νευρωνικών δικτύων για πρόβλεψη είναι ότι είναι σε θέση να μάθουν μόνο από παραδείγματα και ότι μετά την ολοκλήρωση της εκμάθησής τους, είναι σε θέση να αποκρύψουν κρυφές και εντόνως μη γραμμικές εξαρτήσεις, ακόμη και όταν υπάρχει σημαντικός θόρυβος στην training set.

Από την άλλη πλευρά το μειονέκτημα είναι ότι τα NNs μπορούν να μάθουν την εξάρτηση που ισχύει μόνο σε ορισμένη χρονική περίοδο έχοντας ένα σφάλμα πρόβλεψης το οποίο δεν μπορεί να εκτιμηθεί γενικά, όπως για παράδειγμα στην πρόβλεψη χρονοσειρών.

Το μεγαλύτερο κομμάτι αυτής της εργασίας θα ασχοληθεί με την πρόβλεψη της τροχιάς κινούμενων αντικειμένων με την χρήση Νευρωνικών Δικτύων και πιο συγκεκριμένα τα RNNs και τα LSTMs. Γι αυτό το λόγο σε αυτό το κεφάλαιο θα παραθέσουμε παραδείγματα εργασιών που έχουν ασχοληθεί με αυτό το κομμάτι.

#### 3.3.1 Σχετικές Εργασίες

**[1] Pedestrian's Trajectory Forecast in Public Traffic with Artificial Neural Networks,** (Michael Goldhammer, Konrad Doll, Ulrich Brunsmann, André Gensler, Bernhard Sick)

Το παρόν έγγραφο επικεντρώνεται στην πρόβλεψη της κίνησης πεζών ανθρώπων για βραχυχρόνιες τροχιές μέχρι και 2,5s με σκοπό την εφαρμογή της μεθόδου για την διατήρηση της ασφάλειας σε διάφορες εφαρμογές όπως για παράδειγμα οχήματα στα οποία δεν υπάρχει οδηγός. Παρουσιάζεται μια προσέγγιση unsupervised learning βασισμένη πάνω σε μοντέλα νευρωνικών δικτύων για πρόβλεψη της κίνησης.

Σύμφωνα με την έκθεση της οργάνωσης παγκόσμιας υγείας (World Health Organisation), για το 2013, το 22% των συνολικών θανάτων από τροχαία ατυχήματα το χρόνο είναι πεζοί. Το τρέχον θέμα στον τομέα της έρευνας για την ασφάλεια της κυκλοφορίας είναι το Advanced Driver Assistant Systems (ADAS) για τα Ευφυή οχήματα. Αυτό θα μπορέσει να βοηθήσει στην πρόληψη ατυχημάτων ή στο να Ελαχιστοποιήσει τις τα ατυχήματα αυτά με την έγκαιρη προειδοποίηση του οδηγού ή ακόμη και με άμεση παρέμβαση. Ως εκ τούτου, το όχημα Πρέπει να μπορεί να έχει πλήρη εικόνα για το περιβάλλον μέσα στο οποίο κινείται (οδική γεωμετρία, εμπόδια και άλλα). Αυτά τα δεδομένα θα μπορούν να παρέχουν πληροφορίες για την ανάλυση της κατάστασης και την πρόβλεψη μιας επικείμενης σύγκρουσης. Όπως καταλαβαίνουμε είναι πολύ σημαντική η πρόβλεψη της μελλοντικής θέσης των πεζών οι οποίοι βρίσκονται σε αυτό το περιβάλλον.



Η εξόρυξη της τροχιάς των πεζών ανθρώπων γίνεται μέσω κάμερας τελευταίας τεχνολογίας. Το σημείο το οποίο καθορίζεται ως το κέντρο της τροχιάς είναι το κεφάλι του ανθρώπου για πολλούς λόγους, αλλά ο πιο σημαντικός είναι το ότι πάντα, ή σχεδόν πάντα είναι ορατό και σηματοδοτεί το άτομο. Η πρόβλεψη της κίνησης μπορεί να γίνει είτε μέσω των MPLs (Multilayer Perceptrons) είτε μέσω μια πολυονυμικής προσέγγισης στην οποία χρησιμοποιείται το μοντέλο Constant Velocity Kalman Filter.

### CV Model Kalman Filter

Το μοντέλο που χρησιμοποιήθηκε παίρνει σαν δεδομένο ότι η ταχύτητα είναι σταθερή στο επίπεδο της κίνησης (x,y) ενώ η συντεταγμένη z δεν υπολογίζεται αφού μιλάμε για κίνηση σε ένα μόνο επίπεδο. Έτσι έχουμε το παρακάτω διάνυσμα κατάστασης:

$$X_{CV} = [X, Y, \dot{X}, \dot{Y}]^T$$

Καθώς και τον transition matrix για κάθε χρονικό βήμα  $\Delta t$ :

$$A_{CV} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

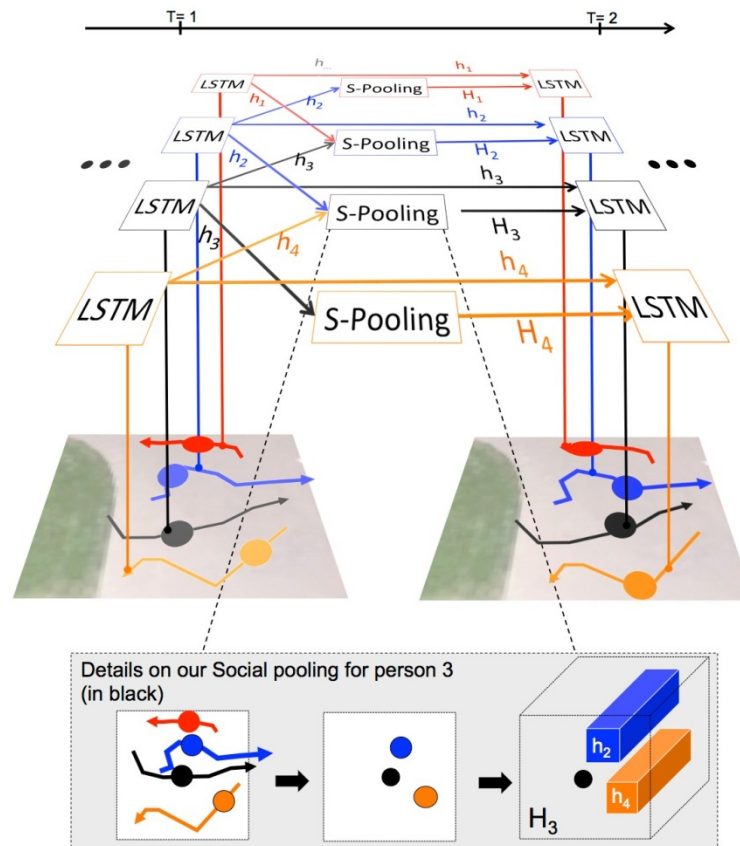
**[2] Social LSTM: Human Trajectory Prediction in Crowded Spaces** (Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, Silvio Savarese Stanford University)

Οι πεζοί ακολουθούν διαφορετικές τροχιές για να αποφύγουν εμπόδια αλλά και να ακολουθήσουν την πορεία κάποιου φίλου τους. Οποιοσδήποτε όχημα που πλοηγείται σε μια τέτοια κατάσταση πρέπει να είναι σε θέση να προβλέψει τις μελλοντικές θέσεις των πεζών και ανάλογα να προσαρμόσει την πορεία του για την αποφυγή συγκρούσεων. Αυτό το πρόβλημα της πρόβλεψης της τροχιάς μπορεί να θεωρηθεί ως έργο δημιουργίας ακολουθιών, όπου μας ενδιαφέρει να προβλέψουμε τη μελλοντική τροχιά των ανθρώπων με βάση τις προηγούμενες θέσεις τους. Μετά την πρόσφατη επιτυχία των μοντέλων RNN για την αλληλουχία πρόβλεψης, προτείνεται ένα μοντέλο LSTM το οποίο μπορεί να μάθει γενικά την ανθρώπινη κίνηση και να προβλέψει τις μελλοντικές τροχιές των πεζών ανθρώπων.

Οι άνθρωποι έχουν την έμφυτη ικανότητα να “διαβάζουν” ουσιαστικά τους ανθρώπους που βρίσκονται στο περιβάλλον τους και να μπορούν να προσαρμόσουν την κίνηση τους ανάλογα με την κατάσταση που επικρατεί γύρω τους. Αυτή η ικανότητα, ενώ φαντάζει εύκολη για έναν άνθρωπο, δεν είναι τόσο εύκολη για μια μηχανή. Μέχρι στιγμής όλες οι μέθοδοι που έχουν χρησιμοποιηθεί για την πρόβλεψη της κίνησης των ανθρώπων βασίζονται σε συναρτήσεις κατασκευασμένες εκ των πρότερων και καμία μέχρι τώρα δεν έχει βασιστεί σε μια data driven προσέγγιση. Με την βοήθεια των LSTMs μπορούμε να συμπεριλάβουμε στην διαδικασία της πρόβλεψης της κίνησης ενός ανθρώπου και την κίνηση άλλων ανθρώπων γύρω του. Έστω ότι για παράδειγμα έχουμε έναν άνθρωπο την χρονική στιγμή  $t$  με συντεταγμένες  $(x_t^i, y_t^i)$ . Παρακολουθούμε την κίνηση όλων των ανθρώπων από την χρονική στιγμή 1 μέχρι την χρονική στιγμή της παρατήρησης  $T_{obs}$  και προβλέπουμε την τροχιά τους για το διάστημα μεταξύ  $T_{obs+1}$  μέχρι και  $T_{pred}$ . Βέβαια ο κάθε άνθρωπός έχει ένα διαφορετικό μοτίβο στην κίνηση του, διαφορετική ταχύτητα, επιτάχυνση κτλ, πράγμα που κάνει την προσέγγιση αυτή πιο δύσκολη με αποτέλεσμα ένα απλό LSTM δίκτυο να μην μπορεί να είναι 100% αποδοτικό.



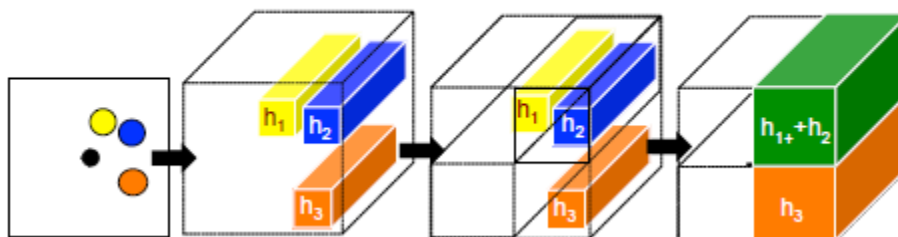
Για αυτό το λόγο το πρόβλημα προσεγγίζεται από μια διαφορετική στρατηγική συγκέντρωσης (pooling strategy) των δεδομένων όπως παρουσιάζεται στην παρακάτω εικόνα.



**Εικόνα 3.3:** Χρησιμοποιείται ένα ξεχωριστό LSTM για κάθε τροχιά σε μια σκηνή. Τα LSTMs στη συνέχεια συνδέονται μεταξύ τους μέσω ενός κοινού (S-pooling) στρώματος. Σε αντίθεση με το παραδοσιακό LSTM, αυτό το στρώμα συγκέντρωσης επιτρέπει σε χωρικά κοντινά LSTMs για να μοιράζονται πληροφορίες μεταξύ τους. Οι κρυφές καταστάσεις των LSTM σε μια συγκεκριμένη ακτίνα συγκεντρώνονται και χρησιμοποιούνται ως είσοδο για το επόμενο βήμα.

Όπως αναφέρθηκε προηγουμένως η τροχιά του κάθε ανθρώπου επηρεάζεται από την κίνηση των γειτόνων του, των οποίων η κίνηση εξαρτάται από τους δικούς τους γείτονες κοκ. Το πρόβλημα το οποίο δημιουργείται είναι ότι ο κάθε άνθρωπος μπορεί να έχει διαφορετικό αριθμό γειτόνων ο οποίος μπορεί να είναι πολύ μεγάλος. Γι αυτό το λόγο χρειάστηκε να δημιουργηθεί ένας τρόπος ο οποίος θα περιέχει όλη την πληροφορία που αναφέρθηκε προηγουμένως σε μι πιο compact μορφή για όλες τις γειτονικές καταστάσεις. Έτσι δημιουργήθηκε το "Social Pooling". Για κάθε χρονική στιγμή κάθε LSTM κελί λαμβάνει συγκεντρωμένη πληροφορία από τα LSTM κελιά των γειτόνων. Κατά την διάρκεια του pooling γίνεται η προσπάθεια να διατηρηθεί η χωρική πληροφορία των γειτόνων του ατόμου από εξετάζουμε όπως φαίνεται στην εικόνα 3.3.





**Εικόνα 3.4:** Παρουσίαση του social pooling για το άτομο που φαίνεται από την μαύρη κουκίδα. Συνδυάζονται οι κρυφές καταστάσεις των γειτόνων (κίτρινο, μπλε και πορτοκαλί) μέσα σε μια συγκεκριμένη χωρική απόσταση. Η συγκέντρωση διατηρεί μερικώς τη χωρική πληροφορία των γειτόνων όπως φαίνεται στα δύο τελευταία βήματα.

Η εκπαίδευση του δικτύου γίνεται με τον ελαχιστοποίηση των χωρικών απωλειών για όλα τα trajectories στο training dataset. Η βασική διαφορά της μεθόδου αυτής με ένα κλασσικό LSTM είναι ότι όλες οι κρυφές καταστάσεις των LSTMs συνοψίζονται στο “Social Pool”.

Στο πείραμα χρησιμοποιήθηκαν 2 ειδών datasets. Το ETH, το οποίο περιέχει δύο scenes, εκ των οποίων η κάθε μια αποτελείται από trajectories 750 ατόμων χωρισμένα σε 2 σετ (ETH και Hotel). Το άλλο dataset αποτελείται και αυτό από 2 scenes με 786 άτομα. Αυτό το dataset αποτελείται από 3 σετ (ZARA 01, ZARA 02 και UCY). Συνολικά η αξιολόγηση του μοντέλου γίνεται από 5 σετ δεδομένων. Τα dataset αποτελούνται από πραγματικά δεδομένα και περιέχουν χιλιάδες trajectories ανθρώπων που βρίσκονται σε πολυσύχναστες περιοχές και δρόμους. Το σφάλμα της εκτίμησης του μοντέλου (prediction error) υπολογίζεται από τα παρακάτω 3 metrics.

1. Average Displacement Error: Το MES (Mean Square Error) των σημείων του trajectory που έχουν προβλεφθεί σε σχέση με τα πραγματικά.
2. Final Displacement Error: Η απόσταση μεταξύ της προβλεπόμενης τελικής κατάστασης και της πραγματικής τελικής κατάστασης στο τέλος της χρονικής στιγμής  $T_{pred}$ .
3. Average non Linear Displacement Error: Το MSE των μη γραμμικών περιοχών του trajectory.

Η εκπαίδευση και αξιολόγηση του μοντέλου γίνεται με τα 4 σετ δεδομένων και το 5ο χρησιμοποιείται σαν test data. Στο τέλος επαναλαμβάνεται όλη η διαδικασία και για τα 5 datasets από την αρχή.

Στην εικόνα 3.5 παρουσιάζεται ένας πίνακας με τα σφάλματα που προαναφέρθηκαν για διαφορετικές τεχνικές εκπαίδευσης και τα διαφορετικά σετ των δεδομένων.



Metric	Methods	Lin	LTA	SF	IGP*	LSTM	our O-LSTM	our Social-LSTM
Avg. disp. error	ETH	0.80	0.54	0.41	<b>0.20</b>	0.60	0.49	0.50
	HOTEL	0.39	0.38	0.25	0.24	0.15	<b>0.09</b>	0.11
	ZARA 1	0.47	0.37	0.40	0.39	0.43	<b>0.22</b>	<b>0.22</b>
	ZARA 2	0.45	0.40	0.40	0.41	0.51	0.28	<b>0.25</b>
	UCY	0.57	0.51	0.48	0.61	0.52	0.35	<b>0.27</b>
	Average	0.53	0.44	0.39	0.37	0.44	0.28	<b>0.27</b>
Avg. non-linear disp. error	ETH	0.95	0.70	0.49	0.39	0.28	<b>0.24</b>	0.25
	HOTEL	0.55	0.49	0.38	0.34	0.09	<b>0.06</b>	0.07
	ZARA 1	0.56	0.39	0.41	0.54	0.24	<b>0.13</b>	<b>0.13</b>
	ZARA 2	0.44	0.41	0.39	0.43	0.30	0.20	<b>0.16</b>
	UCY	0.62	0.57	0.54	0.62	0.31	0.20	<b>0.16</b>
	Average	0.62	0.51	0.44	0.46	0.24	0.17	<b>0.15</b>
Final disp. error	ETH	1.31	0.77	0.59	<b>0.43</b>	1.31	1.06	1.07
	HOTEL	0.55	0.64	0.37	0.37	0.33	<b>0.20</b>	0.23
	ZARA 1	0.89	0.66	0.60	0.39	0.93	<b>0.46</b>	0.48
	ZARA 2	0.91	0.72	0.68	0.42	1.09	0.58	<b>0.50</b>
	UCY	1.14	0.95	0.78	1.82	1.25	0.90	<b>0.77</b>
	Average	0.97	0.74	<b>0.60</b>	0.69	0.98	0.64	0.61

**Εικόνα 3.5:** Οι πρώτες 6 σειρές είναι το Average Displacement Error, οι γραμμές 7 έως 12 είναι το Average non – Linear Displacement Error και οι τελευταίες 6 σειρές είναι το Final Displacement Error. Όλες οι μέθοδοι προβλέπουν τροχιές για μια σταθερή περίοδο 4,8 δευτερολέπτων.

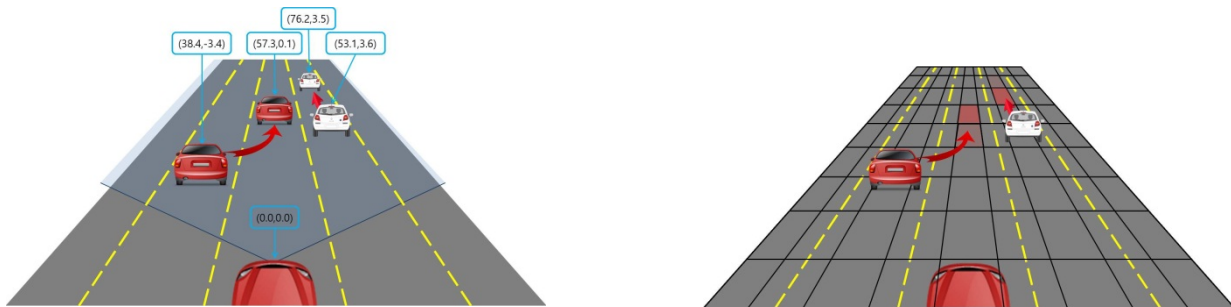
Όπως φαίνεται από τον πίνακα το απλό γραμμικό μοντέλο (Lin) παράγει μεγαλύτερο σφάλμα σε σχέση με τις υπόλοιπες μεθόδους πρόβλεψης, το οποίο είναι μεγαλύτερο στις μη γραμμικές περιοχές. Αντίθετα το μοντέλο το οποίο χρησιμοποιείται σε αυτή την εργασία φαίνεται να έχει τα καλύτερα αποτελέσματα όσον αφορά τα σφάλματα πρόβλεψης. Πιο συγκεκριμένα το λάθος είναι λίγο μεγαλύτερο στο dataset UCY σε σχέση με το ETH dataset αφού το πρώτο περιέχει σημαντικά μεγαλύτερο αριθμό trajectories στις πιο πολυπληθείς περιοχές.

**[3] Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network.** (ByeoungDo Kim, Chang Mook Kang, Seung Hi Lee, Hyunmin Chae, Jaekyum Kim, Chung Choo Chung, and Jun Won Choi Hanyang University, Seoul, Korea)

Σε αυτή την εργασία προτείνεται μια αποτελεσματική μέθοδος πρόβλεψης της τροχιάς κινούμενων οχημάτων με την χρήση των RNNs. Ουσιαστικά τα χαρακτηριστικά της κίνησης ενός οχήματος είναι πολύ διαφορετικά από αυτά των απλών κινούμενων σωμάτων για το λόγο ότι η κίνηση των οχημάτων επηρεάζεται από πολλούς παράγοντες, όπως η δομή του δρόμου, οι κανόνες οδικής ασφάλειας αλλά και την πρόθεση του οδηγού. Άλλες, προηγούμενες μελέτες αποτελούνται από σύνθετους αλγόριθμους λαμβάνοντας υπ όψιν πολλούς παράγοντες για την πρόβλεψη της τροχιάς. Αντίθετα σε αυτό το άρθρο η προσέγγιση είναι πιο απλή αφού βασίζεται σε μοντέλο το οποίο είναι data driven, και η εκμάθησή του γίνεται μέσα από έναν πολύ μεγάλο αριθμό από trajectories, χρησιμοποιώντας deep learning τεχνικές. Η βασική ιδέα της προσέγγισης σε αυτή την εργασία είναι η χρήση ενός LSTM δικτύου για την πρόβλεψη της θέσης αλλά και την κατανόηση της δυναμικής των γύρω οχημάτων. Το LSTM εκπαιδεύεται με σκοπό να μπορεί να προβλέπει την πιθανότητα κατάληψης μιας θέσης, από ένα αντικείμενο, μέσα στον χώρο βασιζόμενο σε προηγούμενες θέσεις.

Οι συντεταγμένες (x, y) ενός οχήματος, όπως φαίνεται στην εικόνα 4.4, είναι οι σχετικές συντεταγμένες θέσης του σε σχέση με αυτές του οχήματος που εξετάζουμε κάθε στιγμή, οι οποίες θεωρούμε ότι είναι οι (0, 0). Έτσι για παράδειγμα οι συντεταγμένες θέσης του ισοτού οχήματος την χρονική στιγμή t θα είναι  $(x_t^{(i)}, y_t^{(i)})$

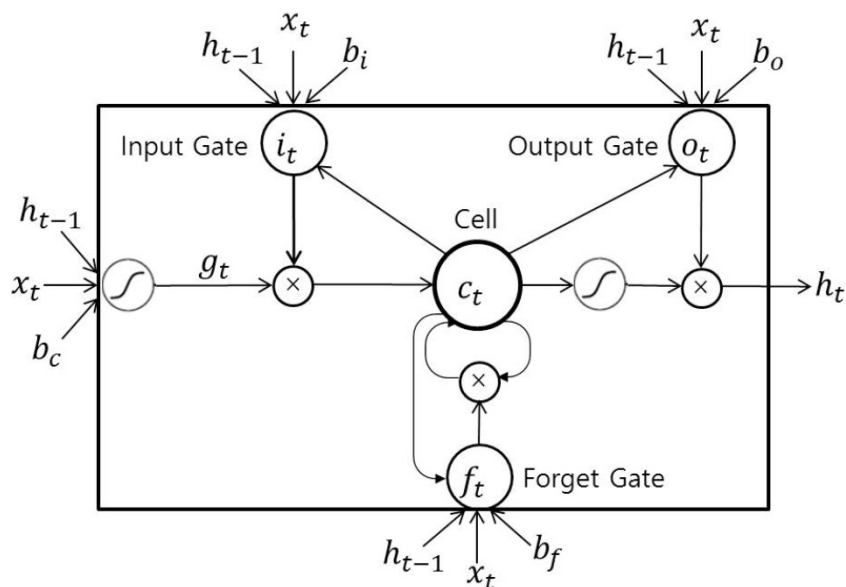




**Εικόνα 3.6:** Αριστερά: Οι σχετικές θέσεις των οχημάτων σε σχέση με το προς εξέταση όχημα. Δεξιά: Μελλοντικές θέσεις των οχημάτων.

Στην ουσία, η βασική λειτουργία του δικτύου είναι να μπορεί να προβλέπει, για την χρονική στιγμή  $t$ , την μελλοντική συντεταγμένη  $(x_{t+\Delta}^{(i)}, y_{t+\Delta}^{(i)})$ , όπου  $\Delta$  ένα χρονικό διάστημα, βασιζόμενη στις προηγούμενες θέσεις του οχήματος. Τα προτεινόμενα αποτελέσματα εκφράζονται σε μορφή πιθανότητας και αποτυπώνονται πάνω στον χώρο, ο οποίος έχει την μορφή πλέγματος, διαστάσεων  $M_x \times M_y$ .

Όπως αναφέραμε και προηγουμένως στο paper αυτό χρησιμοποιείται ένα νευρωνικό δίκτυο τύπου LSTM. Όπως αναφέραμε και στο προηγούμενο κεφάλαιο τα LSTMs αναπτύχθηκαν με σκοπό να ξεπεράσουν το πρόβλημα του vanishing gradient, το οποίο παρατηρείται στα απλά RNNs. Η βασική δομή του δικτύου φαίνεται στην εικόνα 3.7.



**Εικόνα 3.7:** Η βασική δομή του LSTM δικτύου.

Η μνήμη του LSTM “cell” χρησιμοποιείται για να αποθηκεύει τα δεδομένα προηγούμενων χρονικών στιγμών, και μέσω της “Forget Gate” το δίκτυο μπορεί να μαθαίνει “ξεχνάει” παρελθοντικά δεδομένα και να ανανεώνει την κατάσταση του με νέα. Το προς εξέταση όχημα υπολογίζει τις συντεταγμένες των  $N$  κοντινότερων οχημάτων και τις χρησιμοποιεί σαν είσοδο στα  $N$  LSTMs του δικτύου. Κάθε ένα από αυτά τα LSTMs παράγει μια πρόβλεψη θέσης για



κάθε ένα από αυτά τα οχήματα. Σημαντικό να αναφέρουμε είναι ότι από την στιγμή που ο αριθμός των  $N$  κοντινότερων οχημάτων, όπως αυτά προσδιορίζονται από τον αισθητήρα του προς εξέταση οχήματος, εξαρτάται από την οπτική γωνία του αισθητήρα μια πιθανή αλλαγή της κλίσης ίσως προκαλέσει μη αναμενόμενα αποτελέσματα. Για να αποφεύγουν τέτοιου είδους προβλήματα εισάγεται στο μοντέλο η ταχύτητα εκτροπής μαζί με τις συντεταγμένες.

Στο σύστημα ο χώρος καλύπτεται από ένα πλέγμα διαστάσεων  $(M_x, M_y)=(18,11)$  όπως αναφέρθηκε και προηγουμένως. Κάθε στοιχείο του πλέγματος έχει πλάτος 1,75 m και ύψος 10 m. Σε περίπτωση που κάποιο όχημα βγει εκτός των ορίων αυτών τότε χαρακτηρίζεται ως “out of bounds”. Στο πείραμα εξετάζονται τρία σενάρια για διαφορετικές τιμές του  $\Delta$ ,  $\Delta= 0,5$  sec, 1 sec και 2 sec. Το LSTM το οποίο χρησιμοποιείται αποτελείται από 2 layers “cell” μνήμης και ένα layer softmax. Ο αριθμός των κόμβων στο softmax layer είναι  $M_x \cdot M_y + 1 = 199$  (συμπεριλαμβανομένης και της out of boundary κλάσης). Στην Εικόνα 3.8 παρουσιάζεται η ακρίβεια του classification μοντέλου.

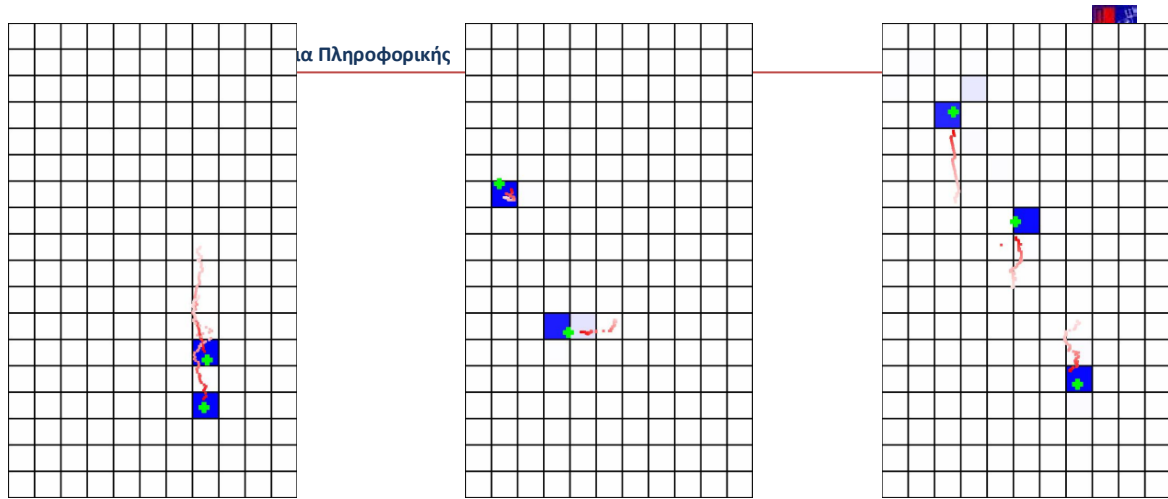
	Prediction term $\Delta$	MAE X (grid)	MAE Y (grid)	MAE (grid)
Proposed Scheme	0.5	0.214	0.271	0.446
	1.0	0.318	0.305	0.558
	2.0	0.707	0.670	1.155
Kalman Filter	0.5	0.266	0.775	0.890
	1.0	0.490	1.510	1.670
	2.0	1.035	2.932	3.214

**Εικόνα 3.8: Σφάλματα πρόβλεψης του μοντέλου σε σχέση με το Kalman Filter \*Weighted Mean Absolute Error–MAE:**

$$MAE = \frac{1}{N} \sum_{i=1}^N \sum_{i_x=1}^{M_x} \sum_{i_y=1}^{M_y} \left\| \begin{bmatrix} i_x \\ i_y \end{bmatrix} - \begin{bmatrix} i_x^{(desired)} \\ i_y^{(desired)} \end{bmatrix} \right\| P_o^{(i)}(i_x, i_y)$$

Όπου  $P_o^{(i)}$  η πιθανότητα κατάληψης της θέσης  $(i_x, i_y)$  στο πλέγμα και  $(i_x^{(desired)}, i_y^{(desired)})$  η θέση του πλέγματος.

Στην εικόνα 3.9 απεικονίζεται ο τρόπος με τον οποίο ο αλγόριθμος προβλέπει την τροχιά για διαφορετικά σενάρια του  $\Delta$ . Οι μικρές κουκίδες με κόκκινο χρώμα αντιπροσωπεύουν την προηγούμενη τροχιά του οχήματος ενώ το πιο σκούρο κόκκινο χρώμα δείχνει πιο πρόσφατες καταστάσεις του ίδιου οχήματος. Η κατάσταση πράσινο χρώμα δείχνει την μελλοντική συντεταγμένη που έχει προβλεφτεί από τον αλγόριθμο. Αξίζει να παρατηρήσουμε πως για μεγαλύτερες τιμές της πιθανότητας έχουμε και πιο σκούρο μπλέ χρώμα για τις μελλοντικές καταστάσεις.



**Εικόνα 3.9: Αποτελέσματα του μοντέλου πρόβλεψης. Αριστερά:  $\Delta=0,5$  sec, Κέντρο:  $\Delta= 1$  sec και Δεξιά:  $\Delta=2$  sec**

#### **[4] A Sparse Recurrent Neural Network for Trajectory Prediction of Atlantic Hurricanes.**

(Mina Moradi Kordmahalleh, Mohammad Gorji Sefidmazgi, and Abdollah Homaifar  
Department of Electrical Engineering, North Carolina A&T State University Greensboro, NC, USA, 27411)

Μια διαφορετική προσέγγιση στην χρήση των RNNs παρουσιάζεται σε αυτό το paper. Με την χρήση των νευρωνικών δικτύων γίνεται μια προσπάθεια για την πρόβλεψη της τροχιάς τυφώνων.

Οι τυφώνες μπορούν να προκαλέσουν μεγάλες φυσικές καταστροφές που οδηγούν στην καταστροφή και απώλεια ζωών. Ως εκ τούτου, για τη μείωση των οικονομικών απωλειών και για να σωθούν ανθρώπινες ζωές, μια ακριβής πρόβλεψη του είναι πολύ σημαντική. Οι δυο βασικές παράμετροι οι οποίες πρέπει να υπολογιστούν και να ληφθούν υπ όψιν για την πρόβλεψη των τυφώνων είναι η τροχιά και η έντασή τους. Η ένταση ενός τυφώνα ορίζεται ως η μέση μέγιστη ταχύτητά του για ένα προκαθορισμένο χρονικό διάστημα, συνήθως 1 έως 10 sec. Σε αυτό το paper χρησιμοποιείται ένα νέο είδος νευρωνικού δικτύου, το sparse RNN, το οποίο δημιουργήθηκε από τους συγγραφείς του παρόντος άρθρου. Η προτεινόμενο νευρωνικό δίκτυο έχει τη δυνατότητα να συσσωρεύσει την πληροφορία που σχετίζεται με τη δυναμική του συστήματος στους κόμβους του δικτύου και να την χρησιμοποιήσει την κατάλληλη στιγμή. Αυτή η ικανότητα καθιστά το sparse RNN δίκτυο κατάλληλο για μοντελοποίηση πολύπλοκων συστημάτων με κρυφές καταστάσεις. Σε αντίθεση με τους απλούς τύπους RNN, εδώ δεν υπάρχουν περιορισμοί στον αριθμό των νευρώνων, στους κόμβους και τα κρυμμένα στρώματα που θα χρησιμοποιηθούν. Επιπλέον, οποιαδήποτε επιπλέον βελτιστοποίηση της δομής του δικτύου και των βαρών σύνδεσης δεν αντιμετωπίζει από ζητήματα αστάθειας, τα οποία μπορεί να υπάρχουν τα κλασικά back propagation νευρωνικά δίκτυα. Έτσι, το προτεινόμενο δίκτυο είναι καταλληλότερο για τη μοντελοποίηση των τυφώνων, ειδικά όταν είναι διαθέσιμες λίγες παρατηρήσεις για την εκπαίδευση του δικτύου, συχνά λόγω της περιορισμένης διάρκειας των τυφώνων. Λόγω αυτού του περιορισμού, για να βρεθούν πιο ακριβείς μοτίβα πρόβλεψης για έναν τυφώνα, το δίκτυο εκπαιδεύεται με δεδομένα από παλαιότερους τυφώνες που στο σύνολό τους έχουν πολλές ομοιότητες και χαρακτηριστικά πολύ κοντινά με τον τυφώνα τον οποίο θέλουμε να προβλέψουμε. Με σκοπό να βρούμε ομοιότητες ανάμεσα σε δυο χρονοσειρές, η επιλογή της συνάρτησης απόστασης είναι πολύ σημαντική. Μια συχνή επιλογή είναι ο υπολογισμός της Ευκλείδειας απόστασης, παρόλο που πολλές φορές τα αποτελέσματα τα οποία παράγει είναι φτωχά από άποψη απόδοσης. Αντίθετα, μια μη γραμμική προσέγγιση, όπως το DTW (Dynamic Time Warping) φαίνεται να λειτουργεί καλύτερα.



### Dynamic Time Warping

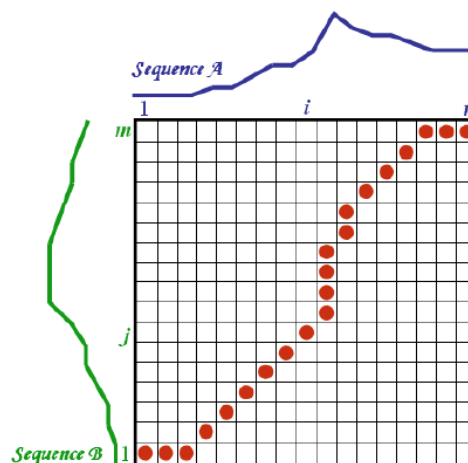
Ας υποθέσουμε ότι έχουμε 2 ακολουθίες A και B όπου:

$$A = a_1, a_2, a_3, \dots, a_n$$

$$B = b_1, b_2, b_3, \dots, b_n$$

Για να βρούμε τις ομοιότητες μεταξύ αυτών των δυο ακολουθιών, μπορούμε να τις ενσωματώσουμε στις άκρες ενός πλέγματος όπως φαίνεται στην εικόνα 4.8 παρακάτω. Τώρα το μόνο που μένει είναι να βρούμε το μονοπάτι μεταξύ των δυο αυτών ακολουθιών που ελαχιστοποιεί την απόστασή τους. Με σκοπό να ελαχιστοποιηθεί ο αριθμός των πιθανών μονοπατιών θα πρέπει να ικανοποιούνται οι παρακάτω συνθήκες.

- Μονοτονία: Το μονοπάτι δεν θα πρέπει να γυρνάει πίσω στον εαυτό του.
- Συνέχεια: Το μονοπάτι θα πρέπει να προχωράει ένα βήμα τη φορά.
- Όριο: Το μονοπάτι θα πρέπει να ξεκινάει από το κάτω αριστερά άκρο και να τελειώνει στο πάνω δεξιά (εικόνα 4.8).
- Παράθυρο στρέβλωσης (Warping Window): Το μονοπάτι δεν θα πρέπει να αποκλίνει πολύ από την διαγώνιο.
- Περιορισμός κλίσης: Το μονοπάτι δεν θα πρέπει να είναι πολύ ρηχό ή πολύ βαθύ.



**Εικόνα 3.10: Κατανομή των ακολουθιών στα δύο άκρα του πλέγματος και υπολογισμός του βέλτιστου μονοπατιού.**

Στην ανάλυση ομοιότητας των δυο ακολουθιών για κάθε ακολουθία  $I$  με γεωγραφικό μήκος  $x^i(t)$  και γεωγραφικό πλάτος  $y^i(t)$  την χρονική στιγμή  $t$  υπολογίζονται οι διαφορές  $\Delta x^i(t)$  και  $\Delta y^i(t)$  όπως φαίνεται παρακάτω

$$\Delta x^i(t) = x^i(t) - x^i(t-1)$$

$$\Delta y^i(t) = y^i(t) - y^i(t-1)$$

Η ταχύτητα και κατεύθυνση του τυφώνα υπολογίζεται από:



$$u_x^i = \frac{\Delta x^i(t)}{\sqrt{\Delta x^i(t)^2 - \Delta y^i(t)^2}} \text{ και } u_y^i = \frac{\Delta y^i(t)}{\sqrt{\Delta x^i(t)^2 - \Delta y^i(t)^2}} \text{ αντίστοιχα.}$$

Χρησιμοποιώντας τις  $T$  πρώτες παρατηρήσεις του τυφώνα  $i$  και του τυφώνα  $j$  η DTW απόσταση υπολογίζεται από τον αλγόριθμο 1 που φαίνεται παρακάτω:

---

**Αλγόριθμος 1: DTW**

---

1. Υπολογισμός του μήκους της χρονοσειράς  $n_p = n_q = T - 1$  και του warping window  $L = T - 1$
  2. Δημιουργία του πλέγματος  $D$  διαστάσεων  $n_p + 1 \times n_q + 1$ .
  3. Αρχικοποίηση των διαστάσεων του πλέγματος  $D = \infty$  και  $D(1,1) = 0$
  4. Για  $l_1 = 1: n_p$  επανέλαβε τα παρακάτω βήματα
  5. Για  $l_2 = \max(l_1 - L, 1) : \min(l_1 + L, n_p)$  επανέλαβε τα παρακάτω βήματα
  6.  $d = \text{norm}(P(l_1, :)) - Q(l_2, :)$
  7.  $D(l_1 + 1, l_2 + 1) = d + \min([D(l_1, l_2 + 1), D(l_1 + 1, l_2), D(l_1, l_2)])$
  8. Επέστρεψε την DTW απόσταση  $D = D(n_p + 1, n_q + 1)$
- 

Στον παραπάνω αλγόριθμο με  $L$  συμβολίζεται το warping window και τα  $P, Q$  περιλαμβάνουν τις κατευθύνσεις των τυφώνων μέσα στον χρόνο.

$$P \triangleq \begin{bmatrix} u_x^i(2) & \dots & u_y^i(2) \\ u_x^i(T) & \dots & u_y^i(T) \end{bmatrix}^T$$

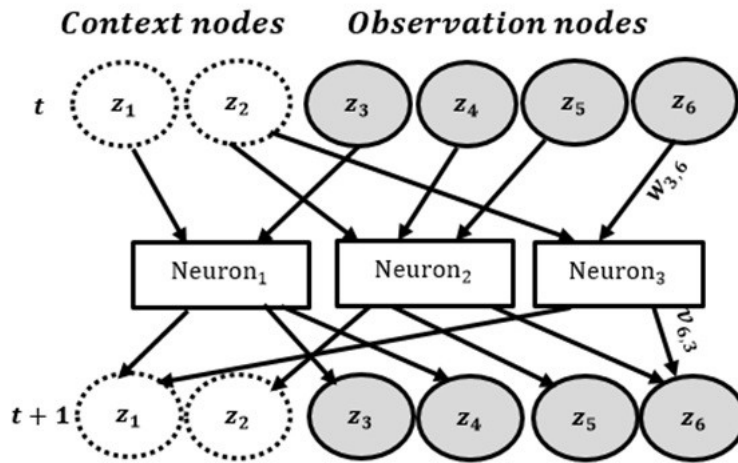
$$Q \triangleq \begin{bmatrix} u_x^j(2) & \dots & u_y^j(2) \\ u_x^j(T) & \dots & u_y^j(T) \end{bmatrix}^T$$

Έχοντας υπολογίσει το DTW μεταξύ του τυφώνα που εξετάζουμε και άλλων τυφώνων μέσα στα δεδομένα υπολογίζεται το score της ομοιότητας από την παρακάτω εξίσωση.

$$S_j = e^{-D}$$

Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκε το 50% από τους καλύτερους, πιο όμοιους τυφώνες, σε σχέση με αυτόν που εξετάζεται.

Το νευρωνικό δίκτυο το οποίο χρησιμοποιείται αποτελείται από μια σειρά από νευρώνες, κόμβους παρατήρησης (observation nodes) και context nodes. Για την πρόβλεψη των τυφώνων, υπάρχουν τέσσερις κόμβοι παρατήρησης οι οποίοι αναπαριστούν τις τιμές των  $\Delta x_i, \Delta y_i, \Delta u_x^i$  και  $\Delta u_y^i$  για τον  $i$  τυφώνα. Οι κόμβοι περιεχομένου (context nodes) δεν περιλαμβάνουν πειραματικές μετρήσεις και παίζουν το ρόλο της μνήμης ή των εσωτερικών καταστάσεων του μοντέλου.



Εικόνα 3.11: Σχηματική αναπαράσταση του Sparse RNN.

Όπως φαίνεται και στην εικόνα 3.11 ο κάθε νευρώνας έχει συνδέσεις εισόδου και εξόδου τόσο από τους κόμβους περιεχομένου όσο και από τους κόμβους παρατήρησης. Για παράδειγμα το νευρωνικό δίκτυο της παραπάνω εικόνας έχει 3εις νευρώνες, ο καθένας από τους οποίους μπορεί να συνδεθεί με 2 context και 4 observation κόμβους. Έτσι οι κόμβοι  $z_1$  και  $z_3$  τροφοδοτούν τον νευρώνα 1 την χρονική στιγμή  $t$  με αντίστοιχα βάρη  $w_{1,1}$ ,  $w_{1,3}$  και η έξοδος του νευρώνα 1 τροφοδοτεί τους κόμβους  $z_1$ ,  $z_3$  και  $z_4$  με αντίστοιχα βάρη  $u_{1,1}, u_{3,1}, u_{4,1}$  για την χρονική στιγμή  $t+1$ .

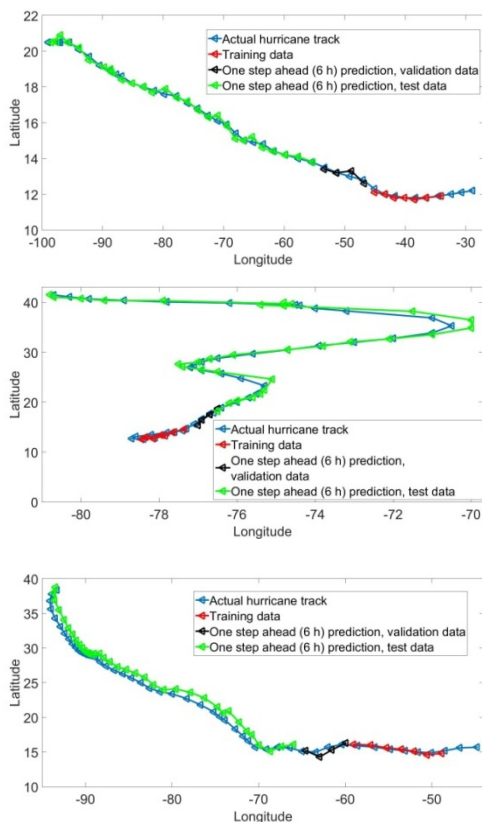
Ο αλγόριθμος ξεκινάει παίρνοντας ένα σετ από τυχαίες υποψήφιες λύσεις (individuals) που ονομάζονται πληθυσμός (population). Η λύση βρίσκεται εξελίσσοντας τις υποψήφιες αυτές λύσεις με την βοήθεια μιας συνάρτησης καταλληλότητας\*. Με την πάροδο του χρόνου βρίσκονται νέες υποψήφιες λύσεις εφαρμόζοντας κατάλληλους τελεστές. Η διαδικασία συνεχίζεται μέχρι να ικανοποιούνται συγκεκριμένες συνθήκες.

$$*F = \frac{1}{T-2} \sum_{i=1}^I \sum_{t=3}^T S_i (Y^i(t) - \hat{Y}^i(t))^2 + \frac{1}{T-3} \sum_{i=1}^I \sum_{t=3}^T S_i (Y^i(t+1) - \hat{Y}^i(t+1))^2 + \dots$$

$$+ \frac{1}{T-m-1} \sum_{i=1}^I \sum_{t=3}^T S_i (Y^i(t+m-1) - \hat{Y}^i(t+m-1))^2 + c$$

Όπου  $I$  είναι ο αριθμός των πιο όμοιων τυφώνων με αυτόν που εξετάζεται,  $S_i$  είναι το σκορ ομοιότητας (similarity score) και  $c$  το άθροισμα των απόλυτων τιμών των βαρών των εισερχομένων και εξερχομένων συνδέσεων του υποψήφιου δικτύου.

Στην εικόνα 3.12 που ακολουθεί παρουσιάζονται συνοπτικά αποτελέσματα του αλγόριθμου για διάφορους γνωστούς τυφώνες με πολύ καλά αποτελέσματα.



Εικόνα 3.12

Πάνω: Πρόβλεψη της τροχιάς του τυφώνα Dean

Κέντρο: Πρόβλεψη της τροχιάς του τυφώνα Sandy

Κάτω: Πρόβλεψη της τροχιάς του τυφώνα Isaac

**[5] Modeling Trajectory as Image: Convolutional Neural Networks for Multi-scale Taxi Trajectory Prediction.** (Jianming Lv, South China University of Technology,Guangzhou, China, Qing Li City University of Hongkong, Xintong Wang South China University of Technology,Guangzhou, China)

Τα ταξί έχουν γίνει σήμερα από τα πιο σημαντικά μέσα μεταφοράς στις μεγάλες πόλεις. Η ακριβής πρόβλεψη του προορισμού της τροχιάς ενός ταξί μπορεί να ωφελησει πολύ τον προγραμματισμό των δρομολογίων. Σε αυτό το κείμενο, προτείνεται ο αλγόριθμος T-CONV, ένας νέος αλγόριθμος πρόβλεψης τροχιάς, ο οποίος διαμορφώνει και απεικονίζει τις τροχίες ως δισδιάστατες εικόνες και χρησιμοποιούν πολλαπλά στρώματα CNNs (Convolutional Neural Networks) για την εξαγωγή πολλαπλών μοτίβων κίνησης με υψηλή ακρίβεια πρόβλεψης. Σε σύγκριση με τους παραδοσιακούς αλγόριθμους που επεξεργάζονται τις τροχίες ως μονοδιάστατες ακολουθίες χωρικών σημείων, T-CONV είναι ευκολότερο να συλλάβει χωρικά χαρακτηριστικά δύο διαστάσεων σε διάφορες κλίμακες. Οι κύριες συνεισφορές αυτού του paper συνοψίζονται παρακάτω:

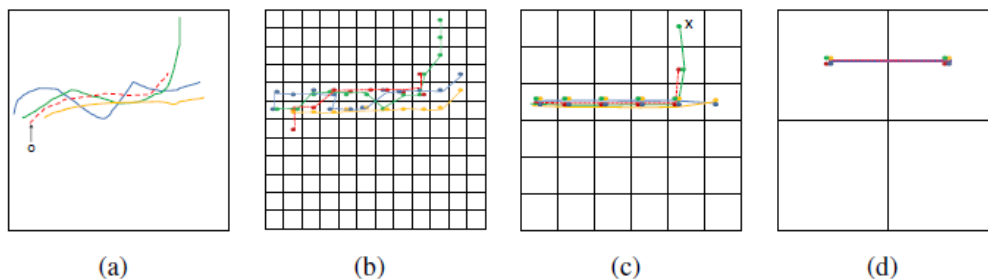
- Διαφορά με τις κλασσικές μεθόδους πρόβλεψης τροχιάς αφού εδώ οι δύο διαστάσεις ενός trajectory μοντελοποιούνται σαν εικόνα και το κάθε σημείο του, απεικονίζεται σαν ένα pixel.
- Είναι η πρώτη φορά που χρησιμοποιούνται CNNs για τον συνδυασμό πολλαπλών χαρακτηριστικών μια τροχιάς με σκοπό την καλύτερη πρόβλεψή της.
- Προκειμένου να επιλυθεί το πρόβλημα της ασυμφωνίας των τροχιών, ενσωματώνονται πολλαπλά τοπικά συνθετικά πεδία στο T-CONV για να καταγράψουν σημαντικές περιοχές στα trajectories. Σε σύγκριση με τα παραδοσιακά CNNs που εφαρμόζεται στην επεξεργασία εικόνων η οποία εκτελεί συνελίξεις σε παγκόσμιες εικόνες, το T-CONV επιλέγει πολλαπλές υποπεριοχές των εικόνων ως πεδία συνελίξης με σκοπό να ξεπεραστεί η ασυμφωνία των δεδομένων.





- Με σκοπό να ελεγχθεί η αποτελεσματικότητα του μοντέλου, χρησιμοποιείται μια άλλη μέθοδος για να απεικονίσει τα μοτίβα τα οποία εξήχθησαν από το T-CONV σε διαφορετικές κλίμακες. Τα πειράματα έδειξαν ότι τα T-CONVs μπορούν να μάθουν τροχιές μικρής κλίμακας στα στρώματα χαμηλού επιπέδου και να τις ενώσουν σε τροχιές μεγάλης κλίμακας στα στρώματα υψηλότερου επιπέδου.]

Στην εικόνα 4.11 που ακολουθεί παρουσιάζονται παραδείγματα από τροχιές ταξί σε διαφορετικές κλίμακες. Στην εικόνα (a) απεικονίζονται τα αρχικά trajectories. Η διακεκομμένη γραμμή απεικονίζει την τροχιά που θέλουμε να προβλέψουμε με αρχικό σημείο το σημείο 0. Στην εικόνα (b) απεικονίζονται οι τροχιές πάνω στο χωρικό πλέγμα όπου φαίνονται με μεγαλύτερη λεπτομέρεια. Με αυτό τον τρόπο μειώνεται σε σημαντικό βαθμό η επικάλυψη των σημείων των διαφορετικών τροχιών και παρουσιάζονται με μεγαλύτερη λεπτομέρεια. Στην εικόνα (c) φαίνονται οι τροχιές σε μεγαλύτερη χωρική κλίμακα με αποτέλεσμα να μπορεί κανείς να καταλάβει την συνολική πορεία και την τάση της τροχιάς, και τέλος στην εικόνα (d) παρουσιάζονται όλα τα trajectories σε μεγαλύτερη κλίμακα όπου όπως φαίνεται έχουμε αλληλοεπικάλυψη όλων των σημείων.



Εικόνα 3.13: παραδείγματα από τροχιές ταξί σε διαφορετικές κλίμακες.

### Το Μοντέλο T-CONV

#### Μοντελοποίηση της τροχιάς μέσω εικόνας

Οι περισσότεροι αλγόριθμοι πρόβλεψης τροχιάς μοντελοποιούν το trajectory στην μορφή

$$T_k = \{P_{k1}, P_{k2}, \dots\}$$

όπου  $P_{ki}$  ( $i \geq 1$ ) ένα σημείο του trajectory. Αυτού του τύπου δεδομένα είναι μονοδιάστατα και είναι αρκετές φορές δύσκολο να προβλέψουν αλλαγές δύο διαστάσεων σε μια κίνηση, όπως στροφές, γωνίες κτλ, στοιχεία που όμως είναι πολύ σημαντικά στο να μπορούμε να έχουμε μια ασφαλή και ακριβής πρόβλεψη. Για να μοντελοποιηθεί η τροχιά σε εικόνα αρχικά δημιουργείται ένα grid διαστάσεων  $M \times M$  όπου  $M$  η μέγιστη διακριτική ικανότητα που θέλουμε να έχει η εικόνα. Κάθε σημείο της τροχιάς αντιστοιχίζεται σε ένα κελί  $C_{mn}$  ( $1 \leq m, n \leq M$ ) στο πλέγμα. Με αυτό τον τρόπο δημιουργείται μια δισδιάστατη εικόνα όπου για κάθε τιμή ενός pixel  $I_k(m, n)$  ( $1 \leq m, n \leq M$ ) αντιστοιχεί στο κελί  $C_{mn}$  του πλέγματος. Έτσι το πρόβλημα ανάγεται στην παρακάτω μορφή: Δεδομένης μια εικόνας  $I_k$  της τροχιάς ενός ταξί που ξεκινάει από το σημείο  $P_{k1}$  πρέπει να προβλεφτεί το γεωγραφικό μήκος και πλάτος της τροχιάς.





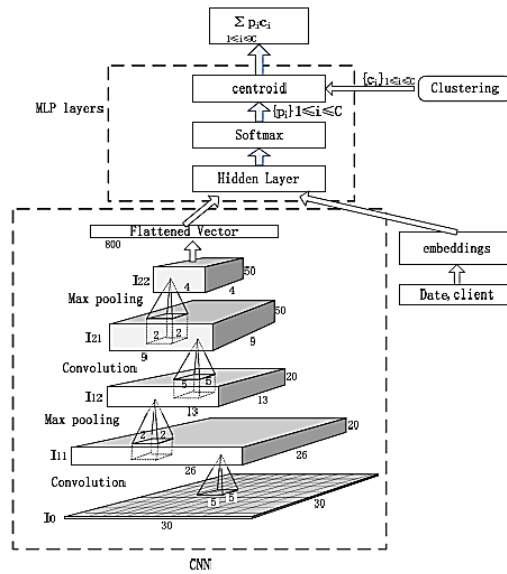
CNNs για την πρόβλεψη τροχιάς

Οι αλγόριθμοι βασισμένοι στα CNNs αναπτύσσονται ευρέως και χρησιμοποιούνται σε μεγάλο βαθμό για classification μοντέλα σε εικόνες. Το CNN μπορεί να εκπαιδευτεί για να εξαγάγει χαρακτηριστικά πολλαπλής κλίμακας από pixels ακατέργαστων εικόνων. Έτσι, μπορούμε να μεταφέρουμε την επιτυχή ανάπτυξη του CNN στις εικόνες για την επεξεργασία εικόνων στην πρόβλεψη τροχιάς. Το T-CONV μοντέλο δέχεται σαν είσοδο μια  $M * M$  εικόνα του trajectory. Δεδομένης μιας τροχιάς ταξί  $T_k = \{P_{k1}, P_{k2}, \dots\}$  η δομή του κάθε pixel  $I_0(m, n)$  της εικόνας μπορεί να πάρει τις παρακάτω τιμές:

$$I_0(m, n) = \left\{ \begin{array}{ll} 0 & \text{if } \exists i(1 \leq i \leq t \wedge P_{ki} \in C_{mn}) \\ 1 & \text{if } P_{ki} \in C_{mn} \\ 0,5 & \text{αλλιώς} \end{array} \right\}$$

Όπως φαίνεται από την παραπάνω εξίσωση  $I_0(m, n)$  παίρνει την τιμή 0 όταν υπάρχει overlapping μεταξύ των σημείων των trajectories αλλιώς παίρνει την τιμή 1, πιο συγκεκριμένα παίρνει την τιμή 1 όταν το τελευταίο σημείο της τροχιάς βρίσκεται μέσα σε ένα κελί του πλέγματος ή αλλιώς παίρνει την τιμή 0,5.

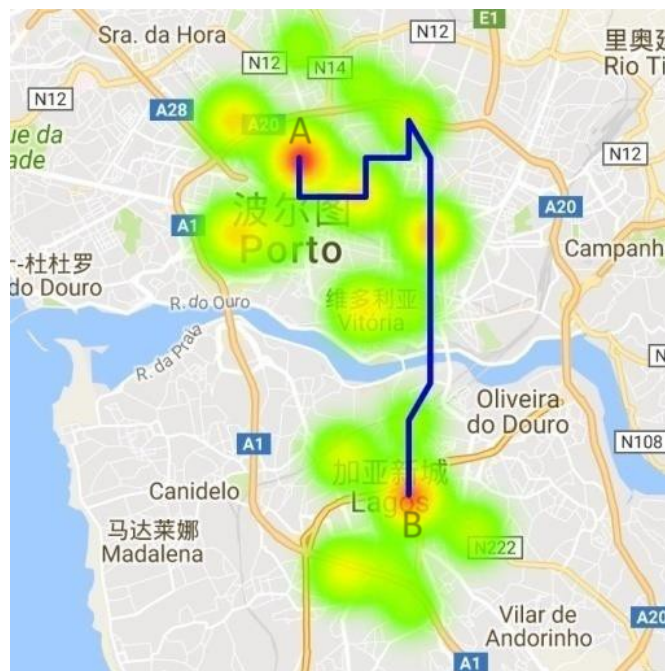
Στην εικόνα 4.12 μπορούμε να δούμε ότι οι διαδικασίες συνέλιξης πραγματοποιούνται πάνω στο  $I_0(m, n)$  με σκοπό να δημιουργηθούν χαρακτηριστικά ανώτερου επιπέδου  $I_{11}$ . Κάθε κυλιόμενο παράθυρο  $5 * 5$  πάνω στο  $I_0(m, n)$  αντικατοπτρίζει μία τιμή στο  $I_{11}$  με αποτέλεσμα να έχουμε στο τέλος της διαδικασίας έναν  $26 * 26$  πίνακα. Στην συνέχεια γίνεται ένα down sampling των χαρακτηριστικών του  $I_{11}$  με σκοπό να αποκτηθεί ένα μικρότερο δείγμα στο  $I_{12}$  κοκ. Έτσι χαρακτηριστικά υψηλότερου επιπέδου παρουσιάζουν τα χαρακτηριστικά σε μεγαλύτερη κλίμακα, τα οποία υπολογίζονται από το συνδυασμό χαμηλότερων επιπέδων χαρακτηριστικά σε μικρότερη κλίμακα. Με αυτόν τον τρόπο, τα μοτίβα πολλαπλών κλιμάκων ενσωματώνονται στην έξοδο του υψηλότερου επιπέδου CNN. Τα χαρακτηριστικά εξόδου του τελευταίου στρώματος CNN έχουν μετατραπεί σε μονοδιάστατες σειρές και τροφοδοτούνται στα στρώματα MLP. Πιο συγκεκριμένα, ο προβλεπόμενος προορισμός υπολογίζεται βάσει του weighted average (σταθμισμένου μέσου) των  $\{c_i\}$  τα οποία έχουν υπολογιστεί από το clustering των δεδομένων παλαιότερων τροχιών. Το διάνυσμα βάρους  $\{p_i\}$  υπολογίζεται εφαρμόζοντας την συνάρτηση ενεργοποίησης softmax στην έξοδο του κρυφού στρώματος όπως φαίνεται στην εξίσωση



Εικόνα 4.12: Η αρχιτεκτονική του T-CONV μοντέλου.

$$p_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}$$

Με σκοπό να γίνει η απεικόνιση των χαρακτηριστικών που έχουν εξαχθεί από τις τροχιές ακολουθείται η αντίστροφη διαδικασία του pooling που είδαμε προηγουμένως. Η αντίστροφη αυτή διαδικασία (deconvolution) χρησιμοποιεί σαν είσοδο τα χαρακτηριστικά από την διαδικασία του max pooling, με σκοπό να ανακτηθούν τα αρχικά μοτίβα των trajectories. Στην παρακάτω εικόνα φαίνεται το αποτέλεσμα της διαδικασίας αυτής, όπου εμφανίζονται τα μοτίβα τα οποία έχουν δημιουργηθεί από το  $I_{22}$  στρώμα του T-CONV global μοντέλου.



Εικόνα 3.14: Το μοτίβο το οποίο δημιουργήθηκε από το στρώμα  $I_{22}$ . Ο προσανατολισμός ορίζεται από τα σημεία A (σημείο έναρξης) και το τελικό σημείο B.

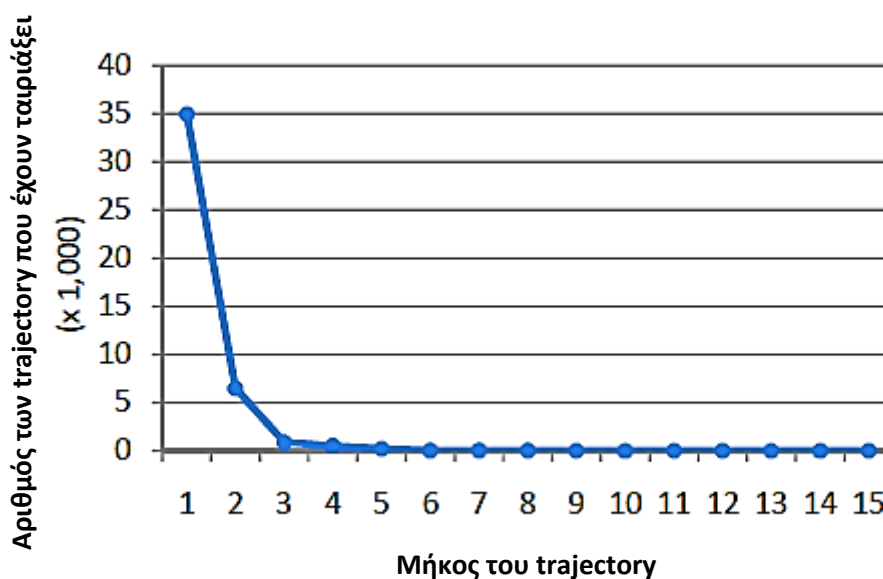


Οι κόκκινες περιοχές στην εικόνα αντικατοπτρίζουν περιοχές που είναι πιο σημαντικές στην τελική πρόβλεψη. Έτσι μπορούμε να δούμε ότι περιοχές κοντά στο σημείο του προορισμού, που είναι πιο σημαντικές, έχουν σημειωθεί με κόκκινο χρώμα.

#### Local CNN με πολλαπλά χωρικά κανάλια

Όπως αναφέρθηκε προηγουμένως, το πρόβλημα της τυχαιότητας των τροχιών είναι συχνό φαινόμενο και μπορεί να έχει σοβαρές παρενέργειες στην ακρίβεια της πρόβλεψης. Στο paper αυτό χρησιμοποιήθηκαν αληθινά δεδομένα από το dataset ECML-PKDD, το οποίο περιέχει περίπου 1,7 εκατομμύρια τροχιές. Ο χάρτης διαχωρίζεται σε ένα πλέγμα  $30 * 30$  και αναπαριστάται κάθε τροχιά ως μία ακολουθία κελιών. Στη συνέχεια επιλέγεται τυχαία μία τροχιά από το σύνολο δεδομένων, και από αυτή επιλέγεται μια υπό-τροχιά, με διαφορετικό μήκος και ελέγχεται ο αριθμός των επικαλυπτόμενων τροχιών στο σύνολο δεδομένων. Αυτό που γίνεται αντιληπτό είναι ότι όταν το μήκος του προς ερώτηση trajectory ξεπερνάει το 3, ο αριθμός των επικαλυπτόμενων τροχιών μειώνεται απότομα. Αυτό δείχνει πως όσο μεγαλύτερο μήκος έχει μια τροχιά τόσο λιγότερες ομοιότητες έχει με άλλες και γι αυτό το λόγο είναι δύσκολο να βρεθούν όμοια μοτίβα στην κίνηση. Αυτό καθιστά δύσκολη να εξόρυξη κοινών προτύπων.

Από την άλλη πλευρά, όπως παρατηρήθηκε από την απεικόνιση του T-CONV global προηγουμένως οι περιοχές κοντά στην αρχή και στο τέλος μιας τροχιάς έχουν πολύ πιο σημαντική συμβολή στην πρόβλεψη του προορισμού από άλλες περιοχές. Αν εφαρμοστεί το μοντέλο μόνο σε αυτές τις σημαντικές τοπικές ζώνες, αντί για όλη την εικόνα, το πρόβλημα της τυχαιότητας των μακροσκελών trajectories μπορεί να ξεπεραστεί.



Εικόνα 3.15: Όσο μεγαλύτερο είναι το μήκος του trajectory τόσο μικρότερη πιθανότητα να υπάρχει συσχέτιση.

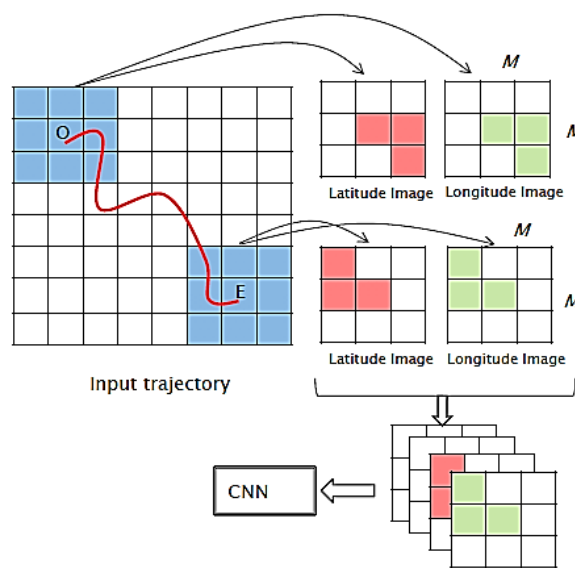
Βάσει της ανάλυσης της προηγούμενης παραγράφου παρουσιάζεται ένα νέο είδους μοντέλο το T-CONV local. Το T-CONV local μοντέλο εστιάζει μόνο στις τοπικές περιοχές της αρχικής και τελικής κατάστασης της τροχιάς, αμελώντας τις υπόλοιπες. Κάθε μια από αυτές τις περιοχές χωρίζεται σε ένα πλέγμα διαστάσεων  $M * M$ . Το γεωγραφικό μήκος και γεωγραφικό πλάτος των σημείων της τροχιάς ενσωματώνεται σε δύο εικόνες: Ia εικόνα



γεωγραφικού πλάτους και  $I_o$  η εικόνα γεωγραφικού μήκους. Έτσι έχοντας ένα trajectory της μορφής  $T_k = \{P_{k1}, P_{k2}, \dots\}$  τα  $I_a$  και  $I_o$  έχουν ως εξής:

$$I_a(m,n) = \begin{cases} f(Lat(C_{mn})) & \text{if } \exists i(1 \leq i \leq t \wedge P_{ki} \in C_{mn}) \\ 0 & \text{αλλιώς} \end{cases}$$

$$I_o(m,n) = \begin{cases} g(Long(C_{mn})) & \text{if } \exists i(1 \leq i \leq t \wedge P_{ki} \in C_{mn}) \\ 0 & \text{αλλιώς} \end{cases}$$



### Τα σφάλματα εκτίμησης για διαφορετικά μοντέλα

Model	error
MLP	2.81
Bidirectional RNN	3.01
Bidirectional RNN with window	2.60
Memory network	2.87
T-CONV-global	2.73
T-CONV-local	2.53

Εικόνα 3.16:

Πάνω: Η αρχιτεκτονική του T-CONV local μοντέλου

Κάτω: Σφάλματα πρόβλεψης για διαφορετικούς τύπους μοντέλων



Η εκτίμηση του σφάλματος στα παραπάνω μοντέλα δόθηκε από την απόσταση Haversine μεταξύ της πραγματικής και προβλεπόμενης τελικής κατάστασης και ορίζεται από τις παρακάτω εξισώσεις:

$$d(x, y) = 2 \cdot r \cdot \arcsin \left( \sqrt{\frac{a}{2}} \right) \text{ όπου}$$

$$a = \sin^2 \left( \frac{\varphi_x - \varphi_y}{2} \right) + \cos \varphi_x \cos \varphi_y \sin^2 \left( \frac{\lambda_x - \lambda_y}{2} \right)$$

Με φ να είναι το γεωγραφικό πλάτος, λ το γεωγραφικό μήκος και r η ακτίνα της Γής.

**[5] Context-Aware Trajectory Prediction in Crowded Spaces.** Federico Bartoli<sup>1</sup>, Giuseppe Lisanti<sup>1</sup>, Lamberto Ballan<sup>1,2</sup>, Alberto Del Bimbo<sup>1</sup> (<sup>1</sup>Media Integration and Communication Center University of Florence, Italy, <sup>2</sup>Department of Mathematics “Tullio Levi-Civita” University of Padova, Italy)

Η ανθρώπινη κίνηση και συμπεριφορά σε πολυσύχναστους χώρους επηρεάζεται από διάφορους παράγοντες, όπως η δυναμική άλλων κινούμενων παραγόντων στην ίδια περιοχή, καθώς και από στατικά στοιχεία που μπορούν να θεωρηθεί ως σημεία έλξης ή εμπόδια. Στην συγκεκριμένη εργασία παρουσιάζεται ένα μοντέλο πρόβλεψης ανθρώπινων τροχιών που είναι ικανό να προβλέψει αλληλεπιδράσεις μεταξύ ανθρώπων μεταξύ τους καθώς και ανθρώπων με χωρικά αντικείμενα. Η πρόβλεψη παράγεται από την παρατήρηση παλιότερων θέσεων των ανθρώπων καθώς και από το ιστορικό των αλληλεπιδράσεων τους με το περιβάλλον τους. Έτσι προτείνεται ένα νευρωνικό δίκτυο “context aware” LSTM το οποίο μπορεί να μάθει και να προβλέπει την κίνηση των ανθρώπων σε πολυσύχναστα μέρη όπως πεζοδρόμια, μουσεία ή ακόμα και σε εμπορικά καταστήματα. Η αξιολόγηση του μοντέλου γίνεται σε πραγματικά δεδομένα, που προέρχονται από βίντεο ανθρώπων που περιφέρονται σε πολυσύχναστα μέρη στον πραγματικό κόσμο όπως για παράδειγμα ένα μουσείο. Τα αποτελέσματα δείχνουν ότι η προσέγγισή αυτή μπορεί να προβλέψει ανθρώπινες τροχιές καλύτερα σε σύγκριση με προηγούμενα πρότυπα μοντέλα.

Οι κύριες συμβολές της παρούσας εργασίας είναι ότι:

- 1) Παρουσιάζεται ένα context aware LSTM μοντέλο το οποίο μπορεί να προβλέψει τόσο την αλληλεπίδραση μεταξύ ανθρώπων, αλλά και μεταξύ ανθρώπου και αντικειμένου.
- 2) Παρουσιάζεται η αποτελεσματικότητα του μοντέλου σε σχέση με άλλες τεχνικές όπως το Social LSTM μοντέλο χρησιμοποιώντας το UCY dataset όπως και ένα νέο, το Museum Visits.



Εικόνα 3.17:

Οι άνθρωποι δεν αλληλεπιδρούν μόνο ο ένας με τον άλλο, αλλά και με τον χώρο που χαρακτηρίζεται από μια πλούσια σημασιολογία.



### Context Aware LSTM

Δεδομένης μια ακολουθίας, κάθε τροχιά εκφράζεται σαν συνάρτηση των δύο συντεταγμένων με την μορφή  $X_t^i(x_t^i, y_t^i)$ . Έτσι χρησιμοποιείται ένα LSTM δίκτυο για την αναπαράσταση της τροχιάς του ιστού ατόμου την χρονική στιγμή  $t$  όπως φαίνεται στις παρακάτω εξισώσεις.

$$\begin{pmatrix} i_t^i \\ f_t^i \\ o_t^i \\ \tilde{c}_t^i \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left[ W \begin{pmatrix} h_{t-1}^i \\ x_t^i \end{pmatrix} + b^i \right]$$

$$x_t^i = \varphi(X_t^i, W_x)$$

$$c_t^i = f_t^i \odot c_{t-1}^i + i_t^i \odot \tilde{c}_t^i$$

$$h_t^i = o_t^i \odot \tanh(c_t^i)$$

όπου  $x_t^i$  είναι η είσοδος του δικτύου,  $h_t^i$  η έξοδος και  $c_t^i$  η κρυφή κατάσταση του δικτύου για την χρονική στιγμή  $t$ . Η είσοδος  $x_t^i$  παράγεται από την εφαρμογή της RELU συνάρτησης  $\varphi(\cdot)$  πάνω στις χωρικές συντεταγμένες και στον πίνακα βαρών  $W_x \in \mathbb{R}^{N+2}$ . Κατά την διαδικασία εκμάθησης (training) του μοντέλου η είσοδοι και έξοδοι από την προηγούμενη χρονική στιγμή ενημερώνονται βάσει του πίνακα των βαρών και του bias. Στην συνέχεια τα διανύσματα αυτά κανονικοποιούνται μέσω της σιγμοειδούς συνάρτησης με αποτέλεσμα τα  $i_t$ ,  $f_t$ ,  $o_t \in \mathbb{R}^D$ , δηλαδή την πύλη εισόδου, την forget πύλη και την πύλη εξόδου αντίστοιχα. Το διάνυσμα  $\tilde{c}_t^i$  απεικονίζει τις νέες υποψήφιες τιμές που μπορούν να προστεθούν στην τελική κατάσταση. Η θέση του ιστού trajectory  $(x, y)_t^i$  υπολογίζεται από την κατάσταση εξόδου  $h_t^i$  και την γκαουσιανή κατανομή την χρονική στιγμή  $t$  μέσω των εξισώσεων:

$$(\mu, \sigma, \rho)_{t+1}^i = \tilde{W} h_t^i$$

$$(x, y)_{t+1}^i \sim N(\mu_{t+1}^i, \sigma_{t+1}^i, \rho_{t+1}^i)$$

Αυτές οι παράμετροι λαμβάνονται από τον γραμμικό μετασχηματισμό της κατάστασης εξόδου  $h_t^i$  με τον πίνακα  $\tilde{W} \in \mathbb{R}^{5 \times D}$ . Η εκμάθηση του δικτύου γίνεται από την ελαχιστοποίηση της συνάρτησης πιθανότητας:

$$L_i(W, W_x, \tilde{W}) = - \sum_{t=1}^{T_{pred}} \log(P((x, y)_t^i | (\mu, \sigma, \rho)_t^i))$$

Παρόλο που αυτά τα δίκτυα είναι ένα πραγματικά ισχυρό εργαλείο για την μοντελοποίηση των χρονικά εξαρτημένων φαινομένων, δεν είναι σε θέση να λάβουν υπόψη τους άλλους παράγοντες που μπορούν να επηρεάσουν την πορεία-τροχιά ενός ατόμου, όπως



αλληλεπιδράσεις με άλλα άτομα και αλληλεπιδράσεις με στατικά στοιχεία του περιβάλλοντος. Μια λύση σε αυτό το πρόβλημα έχει παρουσιαστεί το paper με τίτλο [Alexandre Alahi, Kratharth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In IEEE Conference on Computer Vision and Pattern Recognition, 2016]. Εδώ με σκοπό να εξεταστούν και οι κινήσεις άλλων ατόμων, για κάθε χρονική στιγμή, οι θέσεις τους τοποθετούνται σε ένα πλέγμα διαστάσεων  $m \times n$  στον λεγόμενο occupancy matrix ο οποίος περιγράφεται από την παρακάτω εξίσωση:

$$O_t^i(m, n) = \sum_{j \in G[m, n]} l_{mn} [x_t^i - x_t^j, y_t^i - y_t^j]$$

όπου  $l_{mn}$  η συνάρτηση που επιτρέπει την τοποθέτηση στο κατάλληλο κελί του πλέγματος την αντίστοιχη τροχιά. Αυτό το μοντέλο είναι σε θέση να τροποποιήσει την τροχιά προκειμένου να αποφευχθεί η άμεση σύγκρουση. Ωστόσο, για να έχουμε μια πιο ομαλή πρόβλεψη έχει εισαχθεί ένα δεύτερο μοντέλο, το οποίο λαμβάνει ταυτόχρονα υπόψη τις κρυφές καταστάσεις πολλαπλών LSTM χρησιμοποιώντας ένα pooling στρώμα όπως φαίνεται στην παρακάτω εξίσωση:

$$H_t^i(m, n, :) = \sum_{j \in G_{m \times n}} l_{mn} [x_t^i - x_t^j, y_t^i - y_t^j] Y_{t-1}^j$$

Έχοντας λύσει το πρόβλημα της αλληλεπίδρασης μεταξύ ατόμων μένει να λυθεί και το πρόβλημα της αλληλεπίδρασης ανθρώπου-αντικειμένου στον χώρο. Μια αρχικά απλά προσέγγιση θα ήταν να προσθέσουμε και την τοποθεσία των αντικειμένων αυτών στο μοντέλο και πιο συγκεκριμένα στην γειτονική περιοχή του κάθε ατόμου για κάθε χρονική στιγμή όπως περιγράφεται από την παρακάτω εξίσωση:

$$S_t^i(m, n, k) = \sum_{j \in G_{m \times n}} l_{mn} [x_t^i - p_x^j, y_t^i - p_y^j]$$

Όπου  $(p_x, p_y)^k$  οι συντεταγμένες του αντικειμένου στον χώρο. Αυτή η διαδικασία όμως δεν λαμβάνει υπ όψιν την διαφορετική σημαντικότητα που παρουσιάζει το κάθε αντικείμενο για κάθε άνθρωπο ξεχωριστά, με αποτέλεσμα η πρόβλεψη να είναι αφελής. Γι αυτό το λόγο προτάθηκε η λύση του υπολογισμού της απόστασης του κάθε ατόμου από το κάθε ένα αντικείμενο μέσα στον χώρο για κάθε χρονική στιγμή:

$$C_t^i(k) = \sqrt{(x_t^i - p_x^k)^2 + (y_t^i - p_y^k)^2}$$

Με σκοπό να ενσωματωθεί αυτή η μεταβλητή στο υπόλοιπο μοντέλο, η είσοδος του δικτύου μετασχηματίστηκε σε:

$$x_t^i = [\varphi(X_t^i, W_x) \varphi(C_t^i, W_c)] \text{ και σε}$$

$$x_t^i = [\varphi(X_t^i, W_x) \varphi(C_t^i, W_c) \varphi(O_t^i, W_o)]$$

$$x_t^i = [\varphi(X_t^i, W_x) \varphi(C_t^i, W_c) \varphi(H_t^i, W_H)]$$

ανάλογα με το ποια προσέγγιση αλληλεπίδρασης ανθρώπου -ανθρώπου θέλει να ακολουθήσει κανείς.





Κατά την διάρκεια της φάσης του test του μοντέλου χρησιμοποιήθηκαν trajectories που δεν είχαν χρησιμοποιηθεί στη εκπαίδευση του μοντέλου. Πιο συγκεκριμένα το μοντέλο τροφοδοτήθηκε με τις τοποθεσίες  $\{x_t, y_t\}_1^{t_{obs}}$  όλων των ατόμων που παρατηρήθηκαν στο διάστημα  $(1, t_{obs})$ . Στην συνέχεια υπολογίστηκαν οι τροχιές τους σε χρονικά κοντινές στιγμές  $\{\hat{x}_t, \hat{y}_t\}_{t_{obs+1}}^{t_{pred}}$ , για κάθε άτομο, σε σχέση με:

- Την διαδρομή τους μέχρι την χρονική στιγμή  $t_{obs}$ .
- Την τροχιά άλλων ατόμων την ίδια χρονική περίοδο.
- Τις αποστάσεις σε σχέση με τα στατικά αντικείμενα του χώρου.

Τα δεδομένα που χρησιμοποιήθηκαν στο πείραμα είναι τα UCY και Museum Visits. Το σύνολο δεδομένων UCY αποτελείται από 3 ακολουθίες, τις ZARA-01, ZARA-02 και University. Τα δεδομένα στο ZARA απεικονίζουν περιοχές μεσαίας κίνησης όσον αφορά το πλήθος των ανθρώπων, αντίθετα με το University που είναι δεδομένα περιοχής αυξημένης κίνησης. Όμοια το σύνολο δεδομένων Museum Visits περιέχει μεγάλο αριθμό κινήσεων και ανθρώπων, μόνο που εδώ υπάρχει και η προσθήκη των στατικών αντικειμένων κτλ. Στις παρακάτω εικόνες, παρουσιάζονται οι πίνακες που απεικονίζουν τα σφάλματα πρόβλεψης της μεθόδου αυτής σε σχέση με άλλα γνωστά μοντέλα. Το σφάλμα δίνεται από την εξίσωση του Average Displacement Error το οποίο στην ουσία είναι το Mean Square Error (MSE) της προβλεπόμενης θέσης από την πραγματική:

$$MSE = \sum_{i=1}^N \sum_{t=t_{obs+1}}^{t_{pred}} \frac{\sqrt{(x_t^i - g_t^i)^2 + (y_t^i - g_t^i)^2}}{N(t_{pred} - t_{obs})}$$

Technique	MuseumVisits						UCY (ZARA sequences)		
	Seq1	Seq2	Seq3	Seq4	Seq5	Avg	Seq1	Seq2	Avg
LSTM	<b>0.99</b>	1.22	0.99	0.78	1.03	1.00	1.32	1.49	1.40
O-LSTM	1.60	1.43	0.95	0.76	1.02	1.15	1.65	1.40	1.52
S-LSTM	1.68	1.26	0.94	0.75	0.88	1.10	1.30	1.37	1.34
Context-aware LSTM	1.36	1.14	1.21	<b>0.49</b>	0.82	1.00	1.21	1.37	1.29
Context-aware O-LSTM	1.53	<b>1.08</b>	<b>0.90</b>	0.57	<b>0.80</b>	<b>0.98</b>	<b>1.18</b>	1.34	1.26
Context-aware S-LSTM	1.48	1.27	0.94	0.54	1.07	1.06	1.19	<b>1.25</b>	<b>1.22</b>

Technique	MuseumVisits						UCY (ZARA sequences)		
	Seq1	Seq2	Seq3	Seq4	Seq5	Avg	Seq1	Seq2	Avg
Context-aware LSTM (O)	1.66	1.55	1.26	0.81	1.20	1.30	1.30	1.40	1.35
Context-aware O-LSTM (O)	1.93	1.44	1.32	0.71	1.15	1.31	1.37	1.40	1.39
Context-aware S-LSTM (O)	1.66	1.55	1.26	0.58	1.20	1.25	1.31	1.29	1.30

Εικόνα 3.18:

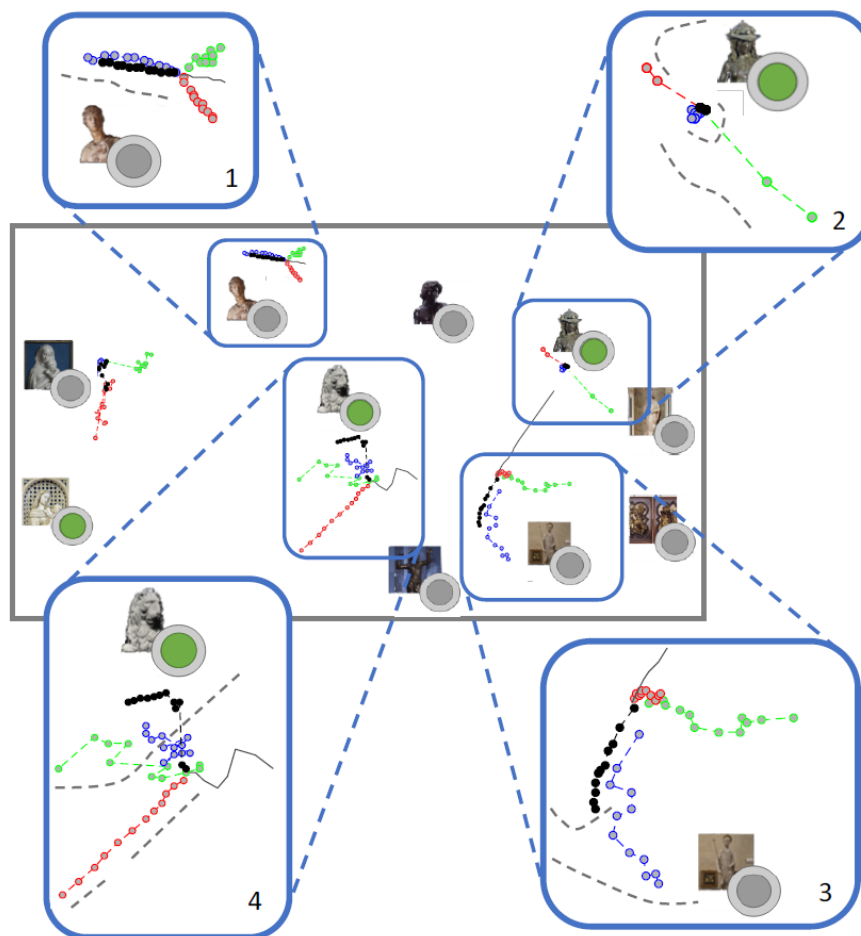
Πίνακας Πάνω: Τα σφάλματα πρόβλεψης στα datasets UCY και Museum Visits

Πίνακας Κάτω: Τα σφάλματα πρόβλεψης στα datasets UCY και Museum Visits με την διαφορετική τεχνική pooling για την αλληλεπίδραση ατόμων με στατικά αντικείμενα.

Τέλος η εικόνα 3.19 δείχνει ορισμένα ποιοτικά αποτελέσματα του μοντέλου σε σύγκριση με προηγούμενες εργασίες. Παρουσιάζονται τέσσερις τροχιές (σε διαφορετικά χρώματα), που λαμβάνονται με διαφορετικές μεθόδους. Με μαύρο, απεικονίζεται η αληθινή τροχιά των ατόμων. Με πράσινο, ένα κλασικό LSTM μοντέλο που δεν χρησιμοποιεί πληροφορίες από τον χώρο και τα άλλα πρόσωπα που δραστηριοποιούνται στη περιοχή, με κόκκινο το



αποτέλεσμα του Social -LSTM, ενώ το μοντέλο που περιγράφεται εμφανίζεται με μπλε χρώμα. Είναι φανερό πως το O-LSTM μοντέλο είναι ικανό να προβλέψει την πραγματική τροχιά χωρίς να επηρεάζεται από το περιβάλλον.



**Εικόνα 4.19:** Ποιοτικά αποτελέσματα, από τα δεδομένα του Museum Visits, που δείχνουν τις προβλεπόμενες τροχιές χρησιμοποιώντας το LSTM (πράσινοι κύκλοι), το Social-LSTM (κόκκινοι κύκλοι), το O-LSTM (μπλε κύκλοι) καθώς και την πραγματική τροχιά (μαύροι κύκλοι). Τέσσερα παραδείγματα που παρουσιάζονται για να συγκριθούν οι τρεις αυτές μέθοδοι. Εμφανίζονται επίσης οι τροχιές άλλων πλησιέστερων ατόμων (Γκριζες διακεκομμένες γραμμές). Ο πράσινος δείκτης καθορίζει ότι ένα έργο τέχνης παρατηρείται από ένα άτομο.

**[6] Soft + Hardwired Attention: An LSTM Framework for Human Trajectory Prediction and Abnormal Event Detection.** Tharindu Fernandoa, Simon Denmana, Sridha Sridharana, Clinton Fookesa

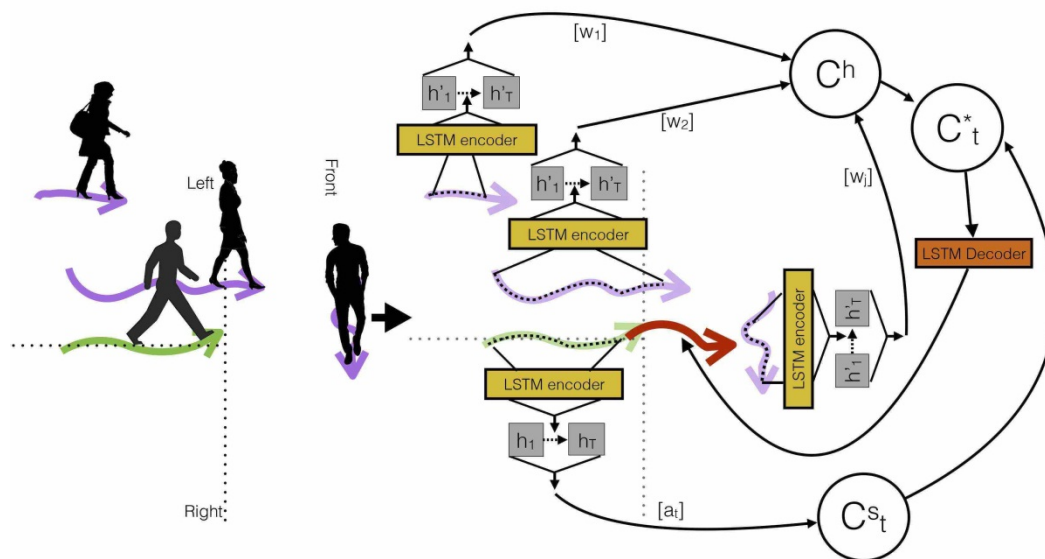
Η κατανόηση και η πρόβλεψη της συμπεριφοράς του πλήθους σε σύνθετα σενάρια πραγματικού κόσμου έχει ένα τεράστιο αριθμό εφαρμογών, από το σχεδιασμό έξυπνων συστημάτων ασφαλείας μέχρι και την ανάπτυξη και δημιουργία κοινωνικά ευαίσθητοποιημένων ρομπότ. Παρά το σημαντικό ενδιαφέρον των ερευνητών σε τομείς όπως ανίχνευση μη φυσιολογικών συμβάντων, εκτίμηση ροής κυκλοφορίας και πρόβλεψη συμπεριφοράς η μοντελοποίηση και η πρόβλεψη της συμπεριφοράς του πλήθους παρέμεινε ένα δύσκολο πρόβλημα λόγω της την πολυπλοκότητά του προβλήματος. Ως άνθρωποι



έχουμε μια διαισθητική ικανότητα πλοήγησης την οποία κατακτούμε μετά από χρόνια πρακτικής και ως εκ τούτου αυτές οι περίπλοκες δυναμικές δεν μπορούν να αποτυπωθούν με λίγα χαρακτηριστικά σε δεδομένα. Πιστεύεται ότι η απευθείας εκμάθηση από τις τροχιές των πεζών που μας ενδιαφέρουν (δηλ. τον πεζό του οποίου η τροχιά είναι αυτή που προσπαθούμε να προβλέψουμε) μαζί με τους γείτονές τους κατέχει το κλειδί για τη μοντελοποίηση της φυσικής ικανότητας των ανθρώπων για πλοήγηση. Η προσέγγιση που παρουσιάζεται σε αυτή την εργασία μπορεί να θεωρηθεί ως μια προσέγγιση που βασίζεται σε δεδομένα που αναπαριστούν τη σχέση μεταξύ των γειτονικών τροχιών σε μια μη επιτηρούμενη μέθοδο εκμάθησης. Η προσέγγιση αυτή οφείλεται στην πρόσφατη επιτυχία των προσεγγίσεων βαθιάς μάθησης με μεθόδους μη επιτηρούμενης μάθησης για εργασίες ταξινόμησης και παλινδρόμησης. Τα δεδομένα του προβλήματος (οι τροχιές δηλαδή) είναι της μορφής

$$x_i = [x_1, y_1, \dots, x_{T_{obs}}, y_{T_{obs}}]$$

Στο συγκεκριμένο πρόβλημα τα σημεία τα οποία θα προβλεφτούν αναφέρονται στις χρονικές στιγμές  $T_{obs+1}$  και  $T_{pred}$  τόσο για το άτομο το οποίο εξετάζουμε όσο και για γειτονικά άτομα τα οποία βρίσκονται στην ίδια σκηνή τις ίδιες χρονικές στιγμές. Η λύση η οποία προτείνεται φαίνεται στην εικόνα 4.18. Για τις ανάγκες κωδικοποίησης και αποκωδικοποίησης χρησιμοποιείται ένα LSTM νευρωνικό δίκτυο και φαίνεται πως η προσέγγιση αυτή είναι ικανή να μάθει και να προβλέψει την ανθρώπινη συμπεριφορά πλοήγησης με επιτυχία.



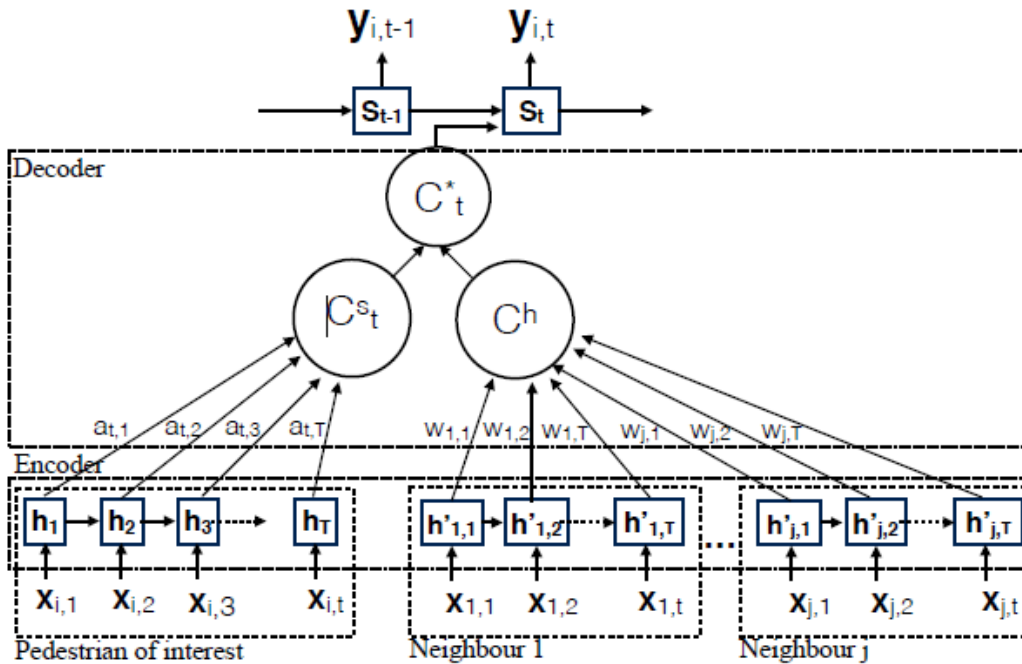
Εικόνα 3.20: Μια σκηνή παρακολούθησης πεζών (στα αριστερά): Εμφανίζεται η τροχιά του ανθρώπου που μας ενδιαφέρει να προβλέψουμε την τροχιά του, με πράσινο χρώμα και έχει δύο γείτονες (που εμφανίζονται με μωβ) στα αριστερά, ένα μπροστά και κανένα δεξιά. Στα δεξιά φαίνεται η σχηματική αναπαράσταση του δικτύου. Οι πληροφορίες τροχιάς κωδικοποιούνται με κωδικοποιητές LSTM. Ένα διάνυσμα  $C_t^s$  χρησιμοποιείται για την κωδικοποίηση και ενσωμάτωση της τροχιάς του ατόμου το οποίο μελετάμε και ένα άλλο διάνυσμα  $C^h$  για τις γειτονικές τροχιές.

Προτεινόμενη προσέγγιση

*Δομή Κωδικοποίησης – Αποκωδικοποίησης LSTM:*



Σε αντίθεση με προηγούμενα μοντέλα τα οποία ευθυγραμμίζουν μια ακολουθία εισόδου με μια ακολουθία εξόδου, εδώ θα ληφθούν υπ όψιν παραπάνω από ένα χαρακτηριστικά τα οποία αφορούν τόσο το υπό μελέτη άτομο όσο και γειτονικά άτομα προς αυτά σε επίπεδο τροχιάς. Το να θεωρήσει κανείς βέβαια ότι όλα τα χαρακτηριστικά που αναφέρθηκαν είναι ίδιας σημασίας για την τροχιά του ατόμου είναι λάθος. Για παράδειγμα ένας άνθρωπος που περπατάει δίπλα στο υπό εξέταση άτομο είναι μεγαλύτερης σημασίας από ένα άτομο που περπατάει κάποια μέτρα μακριά. Βέβαια το πρόβλημα είναι ότι το να τεθούν όλα αυτά τα χαρακτηριστικά σαν είσοδος στο δίκτυο μπορεί να αποδειχθεί εξαιρετικά ενεργοβόρο. Γι αυτό το λόγο και με σκοπό να ξεπεραστεί αυτό το πρόβλημα δημιουργήθηκαν βάρη τα οποία υπολογίζονται βάσει της απόστασης του κάθε πεζού με το υπό εξέταση άτομο.



**Εικόνα 3.21:** Το προτεινόμενο μοντέλο **Soft + Hardwired Attention Model**. Χρησιμοποιούνται οι τροχιές τόσο του ατόμου που εξετάζεται όσο και οι τροχιές των γειτονικών ατόμων. Οι πληροφορίες για την τροχιά του υπό εξέταση ατόμου ενσωματώνεται στο διάνυσμα  $C_t^s$  (soft attention context vector) και οι τροχιές των γειτονικών ατόμων στο διάνυσμα  $C^h$  (hardwired attention context vector). Με σκοπό να παραχθεί η κατάσταση  $C_t^s$  χρησιμοποιείται η συνάρτηση  $a_t$  και τα βάρη σημαίνονται με  $W$ . Με την σύνθεση των δύο αυτών διανυσμάτων παράγεται το τελικό διάνυσμα πρόβλεψης  $C_t^*$

*LSTM Κωδικοποιητής:*

Γενικά ένας κωδικοποιητής λαμβάνει μια ακολουθία  $x$  από την οποία δημιουργεί την κωδικοποιημένη ακολουθία  $h$ . Στην παρούσα εργασία η ακολουθία της τροχιάς δίνεται από την εξίσωση που έχει αναφερθεί σε προηγούμενη παράγραφο και η κωδικοποιημένη ακολουθία δίνεται από την

$$h_i = [h_1, \dots, h_T]$$

με την συνάρτηση κωδικοποίησης να δίνεται από



$$h_t = LSTM [x_t, h_{t-1}]$$

LSTM Αποκωδικοποιητής:

Έχοντας την κρυφή κατάσταση  $s_{t-1}$  για την χρονική στιγμή  $t-1$  και  $y_{t-1}$  την έξοδο του αποκωδικοποιητή για την ίδια χρονική στιγμή ή έξοδος για την χρονική στιγμή  $t$  δίνεται από:

$y_t = f(s_{t-1}, y_{t-1}, C_t)$  όπου  $C_t$  το διάνυσμα περιβάλλοντος (context vector). Το διάνυσμα αυτό εξαρτάται από την είσοδο  $h_i = [h_1, \dots, h_T]$ . Στην προσέγγιση που ακολουθείται σε αυτό το paper το διάνυσμα της τροχιάς του ατόμου  $x_i = [x_1, y_1, \dots, x_{T_{obs}}, y_{T_{obs}}]$  κωδικοποιείται με σκοπό να δημιουργηθεί το διάνυσμα  $C_t^s$  (soft attention context vector) το οποίο υπολογίζεται ως το σταθμισμένο άθροισμα των κρυφών καταστάσεων όπως φαίνεται στην επόμενη εξίσωση:

$$C_t^s = \sum_{j=1}^{T_{obs}} a_{ij} h_j \text{ όπου τα βάρη } a_{ij} \text{ δίνονται από}$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^T \exp(e_{ik})} \text{ και } e_{ij} = a(s_{t-1}, h_j) \text{ όπου η συνάρτηση } a \text{ είναι ένα νευρωνικό δίκτυο}$$

εμπρόσθιας τροφοδότησης για την κοινή εκπαίδευση με άλλες παραμέτρους του συστήματος. Ως επέκταση στο μοντέλο του αποκωδικοποιητή που προτείνεται στο Bahdanau et al. (2014), έχει προστεθεί ένα σύνολο βαρών που χρησιμοποιούνται για την δημιουργία του  $C^h$ . Χρησιμοποιώντας το μοντέλο αυτό συνδυάζουμε τις κωδικοποιημένες κρυφές καταστάσεις των γειτονικών τροχιών στην τοπική γειτονιά. Τα βάρη έχουν σχεδιαστεί για να ενσωματώνουν την έννοια της απόστασης μεταξύ του υπό εξέταση ατόμου και τους γείτονές του στην πρόβλεψη της τροχιάς. Όσο πιο κοντά ένας γειτονικός πεζός, τόσο υψηλότερο είναι το σχετικό βάρος του, έχοντας μεγαλύτερη επιρροή στην τροχιά που προσπαθούμε να προβλέψουμε. Μια απλή αναπαράσταση της συνάρτησης των βαρών δίνεται από

$$w_{(n,j)} = \frac{1}{dist(n,j)} \text{ (hardwired attention weight)}$$

όπου  $dist(n,j)$  η απόσταση μεταξύ του  $n$  γείτονα και αυτού που εξετάζουμε.

Ας υποθέσουμε ότι έχουμε  $N$  γειτονικά trajectories και  $h'(n,j)$  η κωδικοποιημένη κρυφή κατάσταση του  $n$  γείτονα για την χρονική στιγμή  $j$  τότε το βάρος υπολογίζεται ακολούθως:

$$C^h = \sum_{n=1}^N \sum_{j=1}^{T_{obs}} w_{(n,j)} h'_{(n,j)}$$

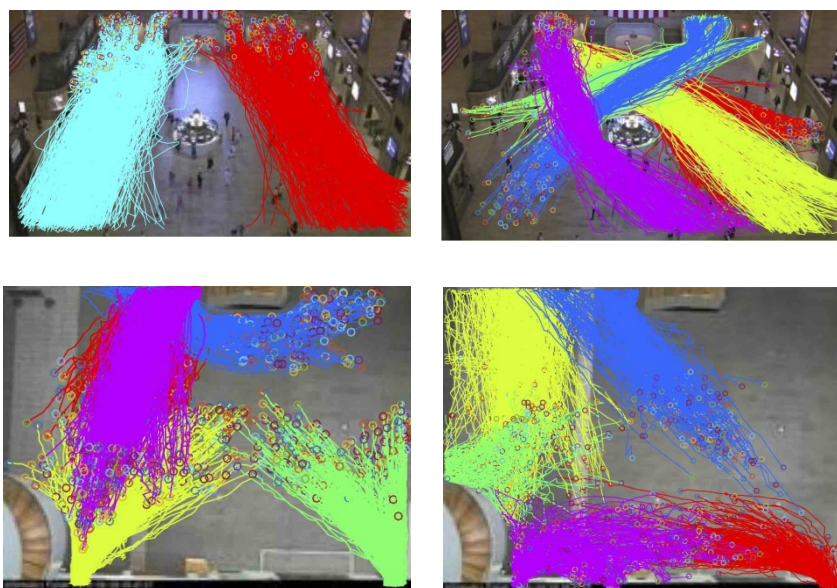
Στη συνέχεια χρησιμοποιούμε ένα απλό στρώμα για να συνδυάσουμε την πληροφορία από τις ατομικές παρατηρήσεις. Κατά συνέπεια το διάνυσμα βάρους (combined context vector) δίνεται από  $C_t^* = \tanh(W_c [C_t^s, C^h])$  όπου  $W_c$  τα βάρη για την συνένωση. Έτσι η τελική πρόβλεψη δίνεται από:



$$y_t = LSTM(s_{t-1}, y_{t-1}, C_t^*)$$

*Πειραματική προσέγγιση:*

Τα δεδομένα τα οποία χρησιμοποιήθηκαν είναι τα New York Grand Central (GC) και Edinburgh Informatics Forum (EIF). Το σύνολο δεδομένων Grand Central αποτελείται από περίπου 12.600 τροχιές, ενώ το Edinburgh Informatics Forum dataset αποτελείται από περίπου 90.000 τροχιές. Για το πρώτο πείραμα στα δεδομένα Grand Central, αφού καθαρίστηκαν τα δεδομένα από μικρές και κατεστραμμένες τροχιές έμειναν 8.000, το μοντέλο εκπαιδεύτηκε με τις 5.000 και αξιολογήθηκε σε 3.000 τροχιές. Στο επόμενο πείραμα εξετάστηκαν τροχιές 3 ημερών από τη βάση δεδομένων Edinburgh Informatics Forum, το μοντέλο εκπαιδεύτηκε σε 10.000 τροχιές και δοκιμάστηκε σε 5.000. Πριν από την εκμάθηση εφαρμόζεται η διαδικασία του clustering με σκοπό να διαχωριστούν οι διαφορετικές συμπεριφορές των ανθρώπων. Για παράδειγμα, ένας πεζός ο οποίος αγοράζει εισιτήρια και άλλοι οι οποίοι εισέρχονται ή εξέρχονται από τον σταθμό του τραίνου. Είναι λογικό πως αυτά διαφορετικά μοτίβα κίνησης μπορούν να δημιουργήσουν μοναδικά, ξεχωριστά και διαφορετικά στυλ τροχιών και μπορούν να αποτυπωθούν σε διαφορετικά μοντέλα. Αυτό μπορεί να επιτευχθεί με το clustering βάση ολόκληρης της τροχιάς, αλλά αυτό θα παράγει πολύ μεγάλο αριθμό ομάδων (clusters) ή μεγάλου αριθμού outliers εξαιτίας της μεγάλης ποικιλίας των διαφορετικών τρόπων της ανθρώπινης κίνησης. Ως αποτέλεσμα, σε κάθε ομάδα, θα έχουμε πολύ λίγα παραδείγματα για να εκπαιδεύσουμε το μοντέλο. Ως λύση στο προαναφερθέν πρόβλημα, η κατηγοριοποίηση (clustering) έγινε μόνο με βάση τις ζώνες εισόδου / εξόδου. Όπως απεικονίζεται στην εικόνα 3.22 αυτή η προσέγγιση λειτουργεί καλά στον διαχωρισμό των διαφορετικών τρόπων ανθρώπινης κίνησης.



**Εικόνα 3.22:** Τα αποτελέσματα του clustering για τα dataset Grand Central (πάνω αριστερά και δεξιά) και για το Edinburgh Informatics Forum (κάτω αριστερά και δεξιά)

Ακόμη και με την ομαδοποίηση με βάση τα σημεία εισόδου και εξόδου, ο τρόπος με τον οποίο ένα άτομο μετακινείται μέσω του περιβάλλοντος και πως επηρεάζεται από τους γείτονές του θα διαφέρει σημαντικά. Για παράδειγμα, μπορούμε να δούμε τις ομάδες που αναπαριστώνται με πράσινο και μπλε χρώμα στην εικόνα 4.20 (πάνω δεξιά). Ο τρόπος που μια τροχιά οδεύει από κάτω προς τα πάνω θα επηρεάσει διαφορετικά τις πράσινες και μπλε τροχιές λόγω των διαφορετικών τους ζωνών εξόδου. Είναι αναμενόμενο και στην ανθρώπινη φύση να μπορεί



και να θέλει να αποτρέψει οποιοδήποτε είδους σύγκρουση κατά την διάρκεια της πορείας του, επομένως, το clustering με βάση τις ζώνες εισόδου / εξόδου αποδεικνύεται πολύ καλό για να καταγράψει αυτή την συμπεριφορά.

*Ποσοτικά αποτελέσματα:*

Σύμφωνα με το Alahi et al. (2016) η ακρίβεια της πρόβλεψης υπολογίζεται βάσει των παρακάτω τριών μετρικών. Αν υποθέσουμε ότι  $n$  είναι ο αριθμός των τροχιών στα δεδομένα δοκιμής (test data),  $x_{i,t}^{pred}$  η πρόβλεψη της θέσης και  $x_{i,t}^{obs}$  η παρατηρούμενη θέση τότε έχουμε:

1. Μέσο Σφάλμα Μετατόπισης (Average Displacement Error – ADE)

$$ADE = \frac{\sum_{i=1}^n \sum_{t=T_{obs}+1}^{T_{pred}} (x_{i,t}^{pred} - x_{i,t}^{obs})^2}{n(T_{pred} - (T_{obs} + 1))}$$

2. Τελικό σφάλμα Μετατόπισης (Final Displacement Error – FDE)

$$FDE = \frac{\sum_{i=1}^n \sqrt{(x_{i,T_{pred}}^{pred} - x_{i,T_{pred}}^{obs})^2}}{n}$$

3. Μέσο σφάλμα μη γραμμικής μετατόπισης (Average non-linear displacement error – n-ADE)

$$n - ADE = \frac{\sum_{i=1}^n \sum_{t=T_{obs}+1}^{T_{pred}} I(x_{i,t}^{pred})(x_{i,t}^{pred} - x_{i,t}^{obs})^2}{\sum_{i=1}^n \sum_{t=T_{obs}+1}^{T_{pred}} I(x_{i,t}^{pred})}$$

όπου

$$I(x_{i,t}^{pred}) = \begin{cases} 1 & \text{if } \frac{d^2 y_{i,t}}{dx^2_{i,t}} \neq 0 \\ 0 & \text{αλλιώς} \end{cases}$$

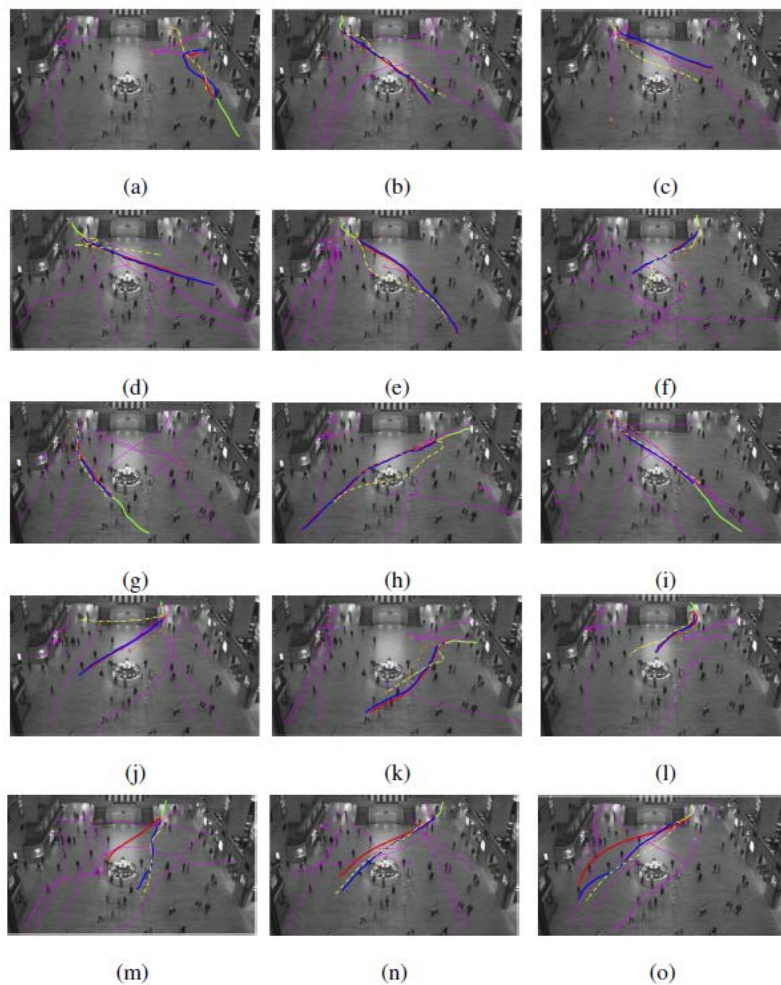
Με σκοπό να αξιολογηθούν τα δυνατά σημεία του προτεινόμενου μοντέλου, συγκρίνεται αυτό το μοντέλο  $OUR_{cmb}$  και δύο παραλλαγές της προτεινόμενης προσέγγισης: 1) Το  $OUR_{sft}$  που αγνοεί τις γειτονικές τροχιές και 2) το  $OUR_{sc}$  που παραλείπει το στάδιο της ομαδοποίησης (clustering) έτσι ώστε μόνο ένα μοντέλο (χρησιμοποιώντας συνδυασμένα βάρη) να μαθένεται.





Metric	Dataset	SF	S-LSTM	OUR <sub>sc</sub>	OUR <sub>sft</sub>	OUR <sub>cmb</sub>
ADE	GC	3.364	1.990	1.878	2.041	<b>1.096</b>
	EIF	3.124	1.524	1.392	1.685	<b>0.986</b>
FDE	GC	5.808	4.519	4.317	5.277	<b>3.011</b>
	EIF	3.909	2.510	2.345	3.089	<b>1.311</b>
n-ADE	GC	3.983	1.781	1.701	2.304	<b>0.985</b>
	EIF	3.394	2.398	2.098	2.415	<b>0.901</b>

Το προτεινόμενο μοντέλο παρουσιάζει πολύ καλύτερα αποτελέσματα από το SF μοντέλο. Η μείωση του λάθους πρόβλεψης είναι πολύ πιο φανερή στα δεδομένα GC το οποίο περιέχει μεγάλο αριθμό ανθρώπων και τροχιών. Συγκρίνοντας τα αποτελέσματα του  $OUR_{sc}$  με το S-LSTM μοντέλο βλέπουμε ότι παρόλη την απουσία της ομαδοποίησης (clustering) υπάρχει μια πολύ καλή πρόβλεψη στην τελική τροχιά. Αντίστοιχα, το μοντέλο  $OUR_{cmb}$  σε σχέση με το  $OUR_{sc}$



Εικόνα 3.23: Ποιοτικά αποτελέσματα: Δοσμένη τροχιά (πράσινο), Πραγματική τροχιά (μπλε), Γειτονικές τροχιές (μοβ) και προβλεπόμενη από το μοντέλο  $OUR_{cmb}$  (με κόκκινο χρώμα), από το μοντέλο S-LSTM (σε κίτρινο χρώμα), από το μοντέλο SF (σε πορτοκάλι). Από το (α) έως (l) ακριβείς προβλέψεις και (m) έως (o) ορισμένες λανθασμένες προβλέψεις.



**Ανίχνευση μη φυσιολογικής συμπεριφοράς:**

Η συγκεκριμένη, προτεινόμενη προσέγγιση του προβλήματος, μπορεί να χρησιμοποιηθεί και για την ανίχνευση ανωμαλιών της κίνησης. Μια απλοϊκή λύση θα ήταν να προβλέψει κανείς την τροχιά για την περίοδο  $T_{obs+1}$  μέχρι και  $T_{pred}$  μετρώντας μόνο τις τιμές για εκείνη την χρονική περίοδο και να μετρήσει την απόκλιση μεταξύ της πραγματικής τιμής και της πρόβλεψης. Έτσι σε περίπτωση που η απόκλιση είναι μεγαλύτερη από ένα κατώφλι, τότε θα μπορούσε να πει κάποιος ότι έχουμε μια ανωμαλία στην κίνηση. Στην πραγματικότητα όμως, λόγω του προσαρμοστικού χαρακτήρα των deep learning δικτύων, τέτοιες συμπεριφορές, όπως για παράδειγμα μια ξαφνική στροφή ή μη φυσιολογικές ταχύτητες, μπορεί να μην κατηγοριοποιηθούν σαν μη φυσιολογικές συμπεριφορές. Γι αυτό το λόγο, παρατηρήθηκε ότι στις κρυφές καταστάσεις του αποκωδικοποιητή/ κωδικοποιητή του LSTM κρατούνται ζωτικής σημασίας πληροφορίες που χρησιμοποιούνται για την μοντελοποίηση της τροχιάς ενός πεζού. Ως εκ τούτου, εάν η συμπεριφορά του είναι ανώμαλη, τότε οι κρυφές τιμές του θα διαφέρουν από εκείνες μιας κανονικής τροχιάς. Έτσι επιλέγηκαν τυχαίας 500 τροχιές από το σύνολο δεδομένων του Grand Central. Οι τροχιές αυτές, διαισθητικά είχαν χαρακτηριστεί σαν μη κανονικές, αφού περιείχαν απότομες στροφές και αλλαγές στην κατεύθυνση της κίνησης. Το σύνολο των δεδομένων περιείχε 445 φυσιολογικά trajectories και 55 μη φυσιολογικά. Οι κρυφές καταστάσεις αυτών των τροχιών περάστηκαν μέσα από τον αλγόριθμο DBSCAN με σκοπό τον εντοπισμό outliers. Παρακάτω παρουσιάζονται τα αποτελέσματα κατηγοριοποίησης μια τροχιάς σαν μη φυσιολογική και με τις δύο προαναφερθείσες μεθόδους.

		Ground Truth	
		Abnormal	Normal
Predicted	Abnormal	<b>47</b>	12
	Normal	8	<b>433</b>
Total		<b>55</b>	<b>445</b>

		Ground Truth	
		Abnormal	Normal
Predicted	Abnormal	<b>29</b>	24
	Normal	26	<b>421</b>
Total		<b>55</b>	<b>445</b>

**Εικόνα 3.24:** Αριστερά: Τα αποτελέσματα κατηγοριοποίησης μη φυσιολογικής τροχιάς από τον αλγόριθμο που προτάθηκε, μέσω των κρυφών καταστάσεων. Δεξιά: Τα αποτελέσματα κατηγοριοποίησης μη φυσιολογικής τροχιάς από την απλοϊκή προσέγγιση που προτάθηκε στην αρχή της παραγράφου.





## **Κεφάλαιο 4 – Υλοποίηση – Τεχνικές Πρόβλεψης**

### **4.1 – Εισαγωγικά**

Στο κεφάλαιο αυτό περιγράφουμε την πρακτική υλοποίηση ενός συστήματος πρόβλεψης τροχιάς ενός αεροπλάνου που χρησιμοποιεί ένα LSTM Neural Network ως προβλέπτη. Έχοντας ως δεδομένη τη κατάσταση του αεροπλάνου την τωρινή στιγμή (π.χ. θέση αεροπλάνου, υψόμετρο, ...) θέλουμε να προβλέψουμε την θέση του αεροπλάνου την επόμενη χρονική στιγμή. Η πρακτική υλοποίηση του συστήματος έγινε με χρήση της γλώσσας προγραμματισμού Python.

### **4.2 – Δεδομένα και Επεξεργασία**

Τα δεδομένα που χρησιμοποιούμε περιέχουν 693 πτήσεις αεροπλάνων από τη Βαρκελώνη στη Μαδρίτη και 703 πτήσεις από τη Μαδρίτη στη Βαρκελώνη. Οι πτήσεις αυτές, περιέχουν δειγματοληψία των ακόλουθων παραμέτρων (σε μη σταθερά χρονικά διαστήματα):

- Η θέση του αεροπλάνου (γεωγραφικό πλάτος και γεωγραφικό μήκος)
- Το υψόμετρο στο οποίο βρίσκεται το αεροπλάνο
- Η θερμοκρασία
- Η ταχύτητα του ανέμου (δύο συνιστώσες - u,v) [2]
- Η υγρασία

Δηλαδή, συνολικά η κατάσταση του αεροπλάνου μια συγκεκριμένη χρονική στιγμή αποτελείται από 7 παραμέτρους. Σκοπός μας είναι η δημιουργία ενός LSTM νευρωνικού δικτύου που με βάση την κατάσταση του αεροπλάνου σε μία συγκεκριμένη μέτρηση (έστω τη στιγμή  $t$ ) που αποτελείται από τις ανωτέρω 7 παραμέτρους, να προβλέπει τη θέση του αεροπλάνου (γεωγραφικό πλάτος και μήκος) στην επόμενη μέτρηση.

Εφόσον τα δεδομένα δεν έχουν σταθερή δειγματοληψία, θα χρειαστεί να τα μετατρέψουμε σε σταθερή δειγματοληψία. Για το σκοπό αυτό χρησιμοποιούμε το εργαλείο “1-pass fixed-rate linear resampler in Matlab/Octave” που υλοποιήθηκε από το “Data Science Lab” του πανεπιστημίου του Πειραιά. Χρησιμοποιώντας το script αυτό μετατρέψουμε τα δεδομένα σε σταθερή δειγματοληψία. Επιπλέον, για να μελετήσουμε το σφάλμα πρόβλεψης για διαφορετική χρονική στιγμή στο μέλλον δημιουργούμε συνολικά μέσω της επεξεργασίας, 10 διαφορετικά σύνολα δεδομένων, με διαφορετικό βήμα σταθερής δειγματοληψίας:

- Κάθε 5 δευτερόλεπτα
- Κάθε 10 δευτερόλεπτα
- Κάθε 30 δευτερόλεπτα
- Κάθε 40 δευτερόλεπτα
- Κάθε 60 δευτερόλεπτα
- Κάθε 120 δευτερόλεπτα
- Κάθε 180 δευτερόλεπτα
- Κάθε 240 δευτερόλεπτα
- Κάθε 300 δευτερόλεπτα
- Κάθε 600 δευτερόλεπτα

Τα αρχεία αυτά βρίσκονται στον φάκελο “datasets” του κώδικα σε μορφή “.csv”.

Στον επόμενο πίνακα βλέπουμε τον αριθμό των συνολικών παραδειγμάτων που αποκτούμε σε κάθε dataset (κάθε παράδειγμα αποτελείται από την τωρινή κατάσταση του αεροπλάνου μαζί με την αντίστοιχη επόμενη θέση του αεροπλάνου που θέλουμε να προβλέψουμε):



Sampling rate (seconds)	Total Examples
5	826825
10	413761
30	138388
60	69546
120	35108
180	23647
240	17904
300	14468
600	7589

#### 4.3 – Αρχιτεκτονική Δικτύου

Το νευρωνικό δίκτυο που θα υλοποιήσουμε χτίστηκε με χρήση της high level python βιβλιοθήκης για νευρωνικά δίκτυα keras που με τη σειρά του χρησιμοποιεί για την κατασκευή του δικτύου είτε την βιβλιοθήκη Theano είτε την βιβλιοθήκη Tensorflow (ανάλογα με την προτίμηση του χρήστη).

Η αρχιτεκτονική του δικτύου που υλοποιήσαμε είναι η ακόλουθη:

- **Είσοδος:** Διάνυσμα μήκους  $l=7$  (όσο και το μήκος των παραμέτρων της κατάστασης του αεροπλάνου) (7 values/timestep)
- **Επίπεδο 1:** LSTM επίπεδο με 16 νευρώνες
- **Επίπεδο 2:** Πλήρως συνδεδεμένο επίπεδο (fully connected) με 8 νευρώνες. Ως συνάρτηση ενεργοποίησης χρησιμοποιούμε τη ReLU.
- **Επίπεδο 3 (Εξοδος):** 2 νευρώνες με σιγμοειδή συνάρτηση ενεργοποίησης. Η έξοδος των νευρώνων αντιστοιχεί στην θέση του αεροπλάνου (γεωγραφικό πλάτος και γεωγραφικό μήκος)

Το μέγεθος των κρυφών επιπέδων τέθηκε έτσι μετά από πειραματισμό με διαφορετικές αρχιτεκτονικές.



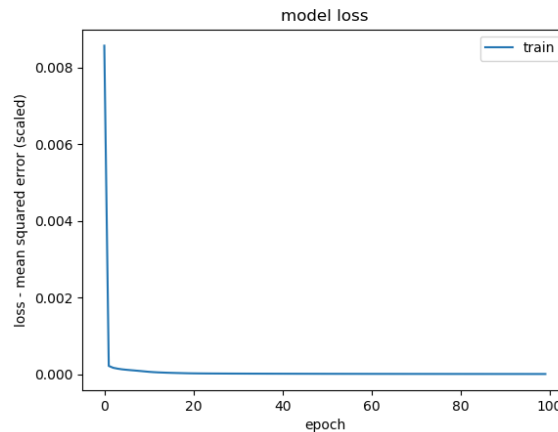
### 4.3.1 – Εκπαίδευση Δικτύου

Χρησιμοποιώντας το αρχείο “aviation\_data\_new.py” φορτώνουμε τα datasets. Για τον έλεγχο του μοντέλου μας χρησιμοποιούμε την μέθοδο Kfold Cross Validation με  $K = 10$ . Με αυτή τη μέθοδο, χωρίζουμε τα παραδείγματα του dataset σε 10 ίσα σύνολα. Στη συνέχεια, τρέχουμε 10 φορές την εκπαίδευση, χρησιμοποιώντας κάθε φορά το ένα από τα σύνολα ως Test set και τα άλλα 9 ως Training set. Επιπλέον, για κάθε διαφορετική από τις 8 παραμέτρους η μέγιστη και η ελάχιστη τιμή της έτσι ώστε να κανονικοποιήσουμε κάθε παράμετρο στο εύρος 0-1. Η κανονικοποίηση αυτή των δεδομένων βοηθάει στην ταχύτερη και πιο εύρωστη σύγκλιση της εξόδου του νευρωνικού δικτύου.

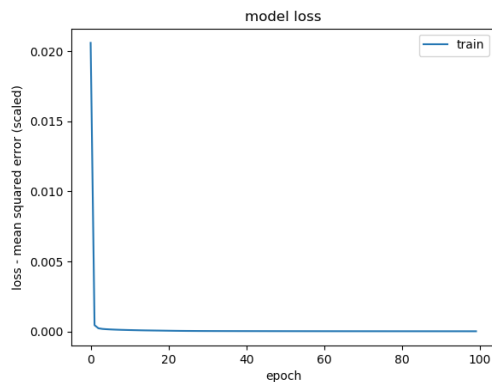
Ως συνάρτηση σφάλματος για την εκπαίδευση του δικτύου χρησιμοποιούμε το μέσο τετραγωνικό σφάλμα μεταξύ της προβλεφθείσας θέσης του αεροπλάνου και της πραγματικής.

Η μέθοδος βελτιστοποίησης που χρησιμοποιούμε είναι η Adam[1] με ρυθμό εκπαίδευσης 0.001. Κατά την εκπαίδευση του δικτύου χρησιμοποιούμε μέγεθος batch 512, δηλαδή κάθε φορά το νευρωνικό εκπαιδεύεται με 512 διαφορετικά παραδείγματα από το σύνολο εκπαίδευσης και εκπαιδεύουμε για συνολικά 100 εποχές.

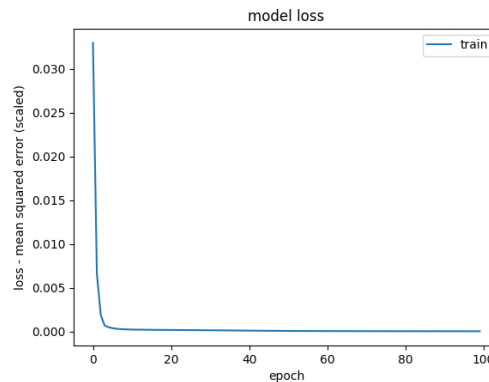
Στα ακόλουθα σχήματα βλέπουμε πως μειώνεται το σφάλμα εκπαίδευσης και επαλήθευσης καθώς εκπαιδεύεται το νευρωνικό δίκτυο, για τα διαφορετικά datasets:



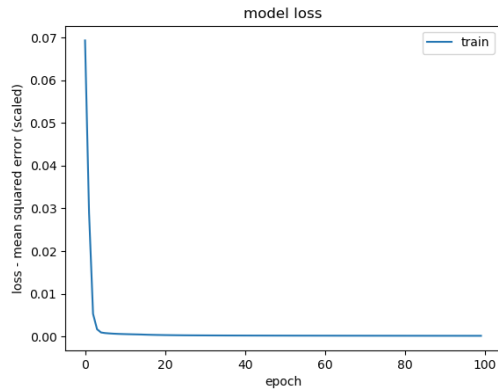
**5 Δευτερόλεπτα**



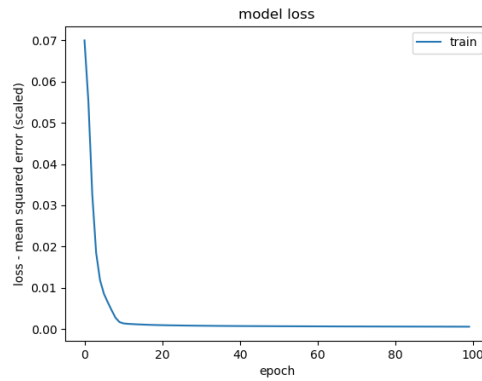
**10 Δευτερόλεπτα**



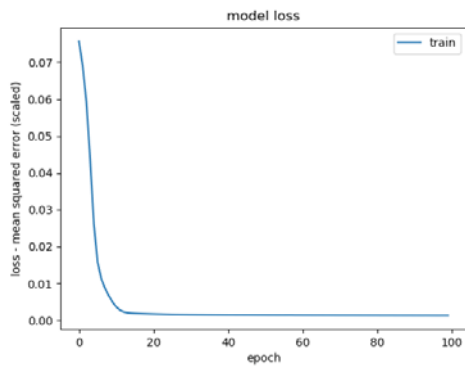
**30 Δευτερόλεπτα**



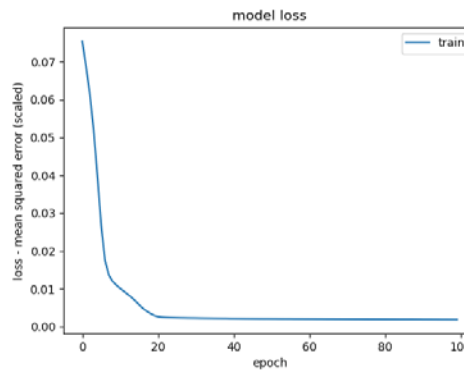
**60 Δευτερόλεπτα**



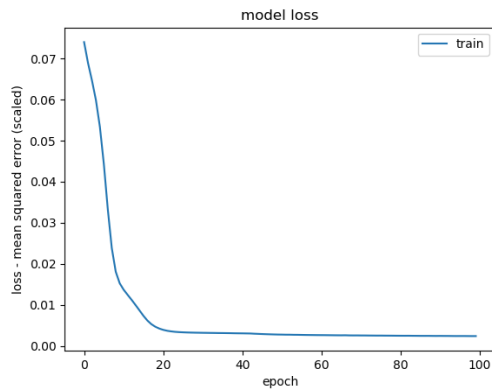
**120 Δευτερόλεπτα**



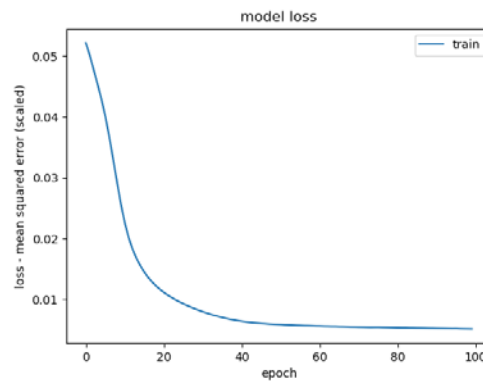
**180 Δευτερόλεπτα**



**240 Δευτερόλεπτα**



**300 Δευτερόλεπτα**



**600 Δευτερόλεπτα**

**Εικόνα 4.1:** Στα παραπάνω σχήματα φαίνεται η εξέλιξη του σφάλματος στο σύνολο εκπαίδευσης κατά τη διάρκεια εκπαίδευσης του νευρωνικού δικτύου. Υπενθυμίζουμε ότι το σφάλμα είναι το μέσο τετραγωνικό σφάλμα μεταξύ της προβλεφθείσας θέσης του αεροπλάνου και της πραγματικής (σε γεωγραφικές συντεταγμένες), και ότι οι θέσεις του αεροπλάνου έχουν υποστεί κανονικοποίηση.



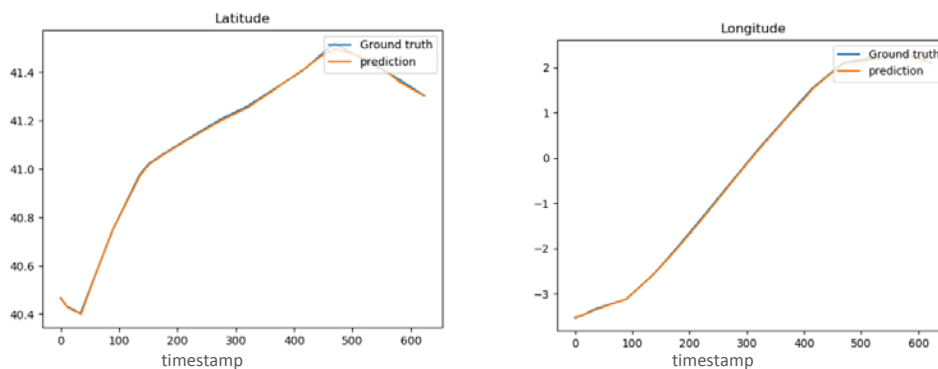
Από τα σχήματα αυτά βλέπουμε ότι όσο πιο μακριά στο μέλλον είναι η επόμενη κατάσταση που θέλουμε να προβλέψουμε, τόσο πιο αργά συγκλίνει και μειώνεται το σφάλμα στο σύνολο εκπαίδευσης (και επίσης, τόσο μεγαλύτερο είναι). Όσον αφορά την ταχύτητα σύγκλισης, μεγάλη σημασία έχει ο αριθμός των παραδειγμάτων που βλέπει σε κάθε εποχή το δίκτυο.

Μετά από την εκπαίδευση του δικτύου, για κάθε διάνυσμα κατάστασης στο σύνολο ελέγχου, προβλέπουμε την επόμενη θέση του αεροπλάνου (γεωγραφικές μοίρες - πλάτος και μήκος) και υπολογίζουμε τη ρίζα του μέσου τετραγωνικού σφάλματος θέσης (RMSE - root mean squared error) (αποκανονικοποιημένο). Στον ακόλουθο πίνακα βλέπουμε το συνολικό σφάλμα που προκύπτει για τα διαφορετικά datasets (όπου σε κάθε dataset έχουμε κάνει cross validation).

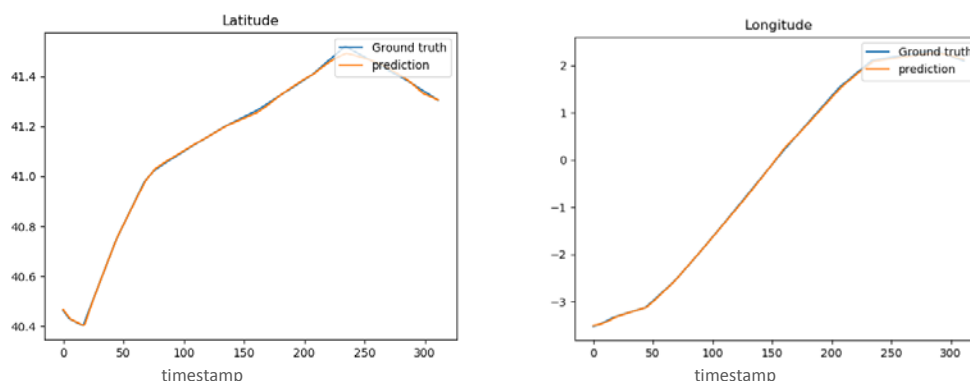
Δευτερόλεπτα Δειγματοληψίας	Σφάλμα Πρόβλεψης (10 Fold CV) (RMSE)
5	0.007
10	0.009
30	0.02
60	0.035
120	0.06
180	0.091
240	0.121
300	0.155
600	0.30

Στα επόμενα σχήματα δείχνουμε επίσης μερικές ενδεικτικές προβλεφθείσες τροχιές από το εκάστοτε test set (και από τα διαφορετικά datasets (διαφορετικές δειγματοληψίες)):

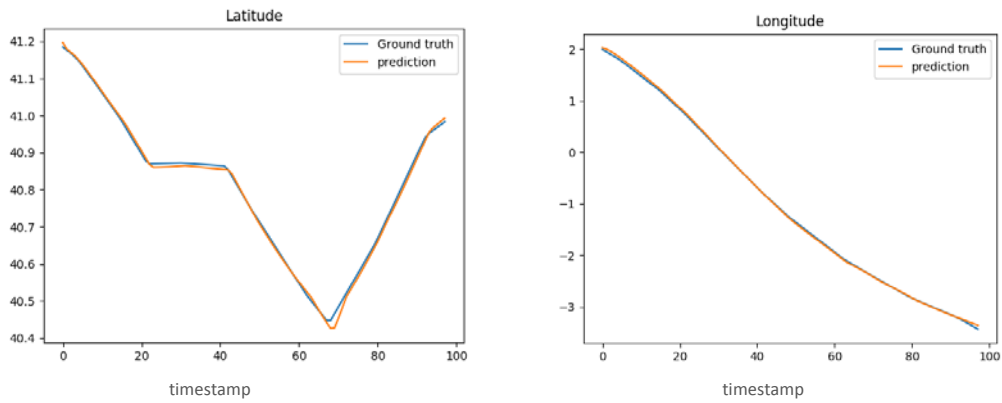
Να σημειωθεί πως κάθε σημείο της προβλεφθείσας τροχιάς χρησιμοποιεί τις μετρήσεις της αληθινής τροχιάς την προηγούμενη στιγμή)



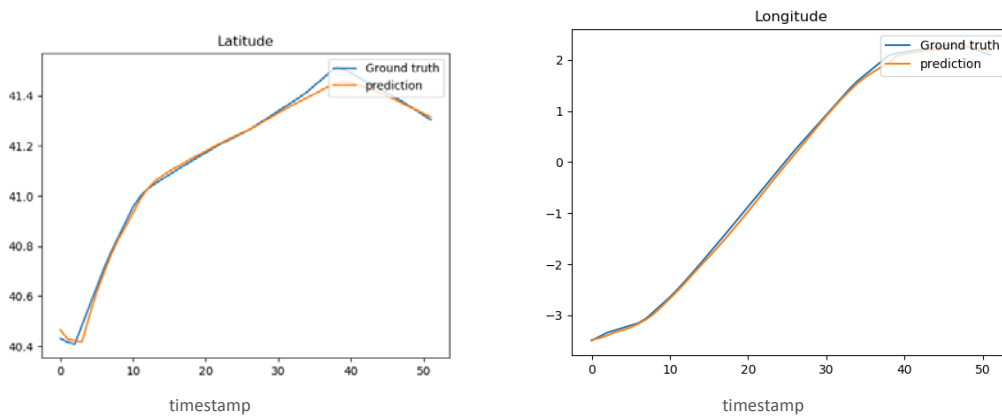
Εικόνα 4.2: Πραγματικές και προβλεφθείσες θέσεις (5 Δευτερόλεπτα).



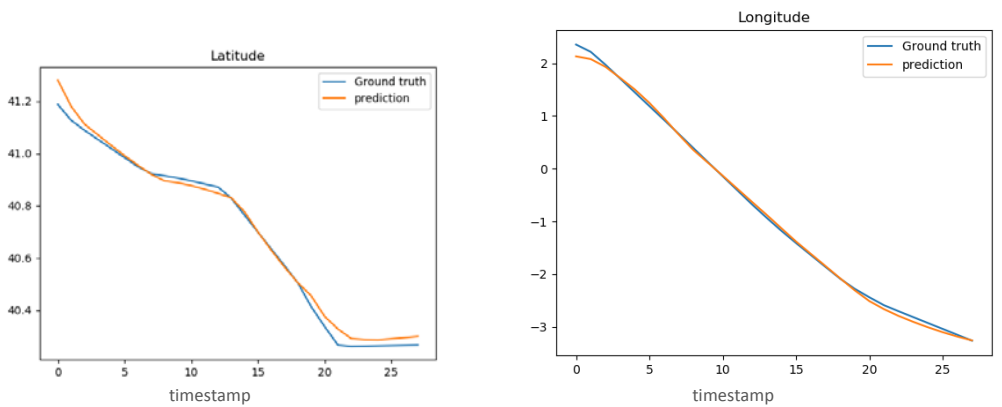
Εικόνα 4.3: Πραγματικές και προβλεφθείσες θέσεις (10 Δευτερόλεπτα).



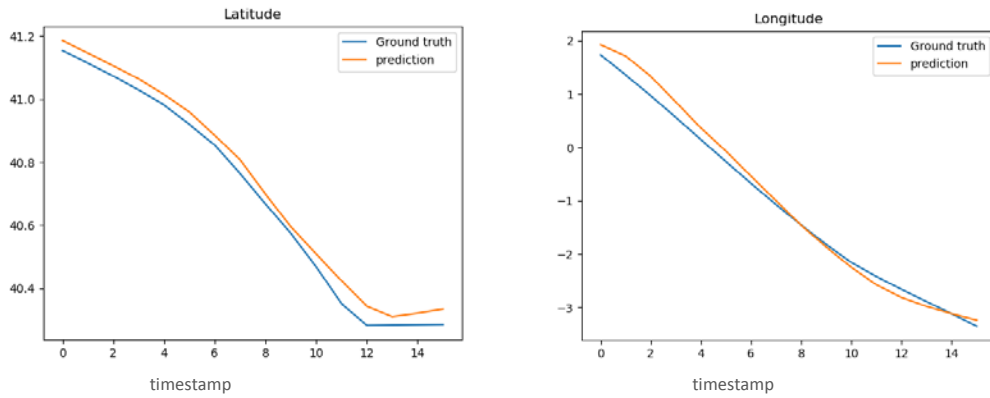
**Εικόνα 4.4: Πραγματικές και προβλεφθείσες θέσεις (30 Δευτερόλεπτα).**



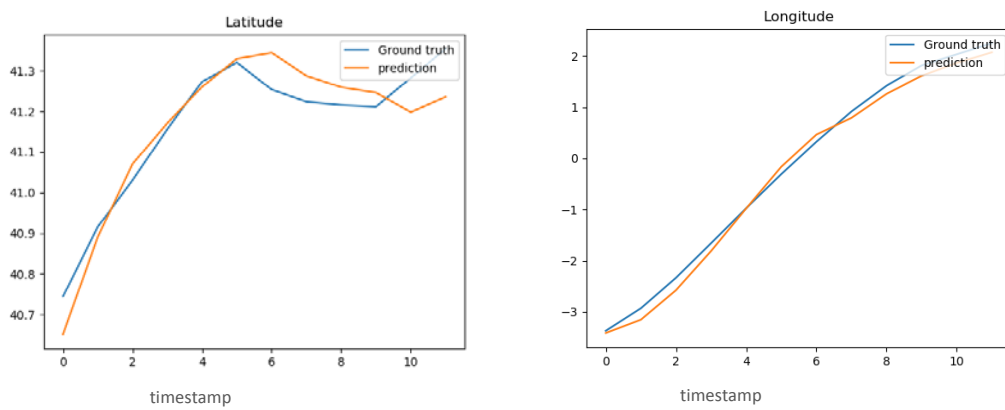
**Εικόνα 4.5: Πραγματικές και προβλεφθείσες θέσεις (60 Δευτερόλεπτα).**



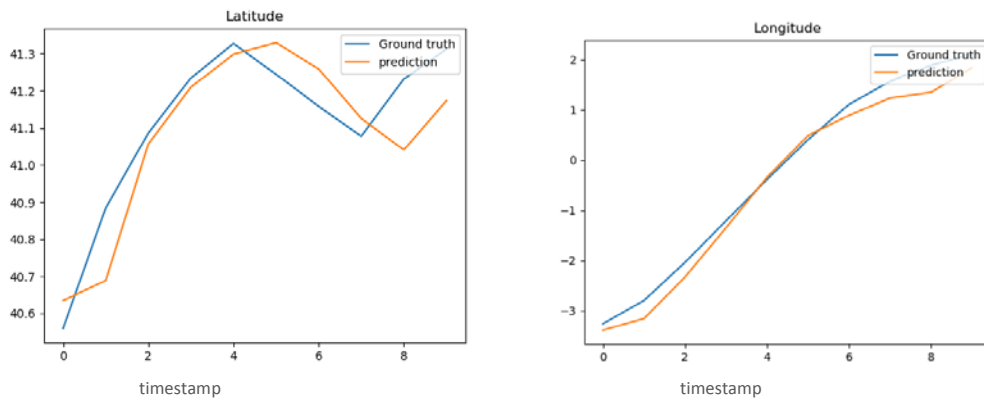
**Εικόνα 4.6: Πραγματικές και προβλεφθείσες θέσεις (120 Δευτερόλεπτα).**



Εικόνα 4.7: Πραγματικές και προβλεφθείσες θέσεις (180 Δευτερόλεπτα).

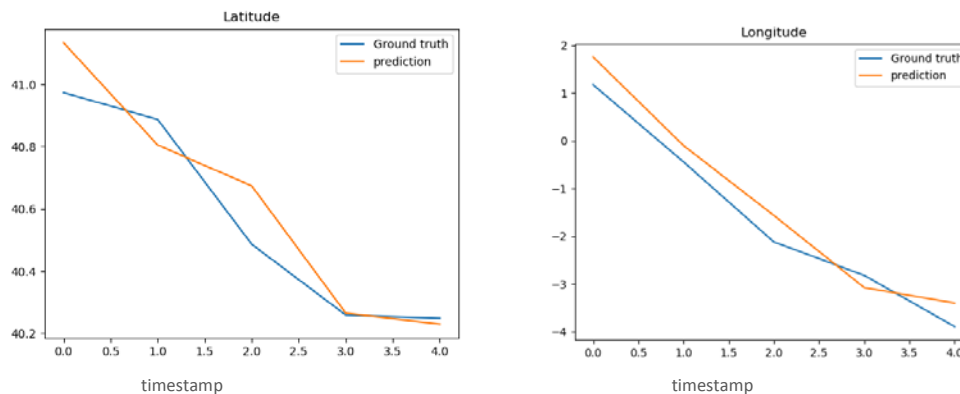


Εικόνα 4.8: Πραγματικές και προβλεφθείσες θέσεις (240 Δευτερόλεπτα).



Εικόνα 4.9: Πραγματικές και προβλεφθείσες θέσεις (300 Δευτερόλεπτα).





**Εικόνα 4.10: Πραγματικές και προβλεφθείσες θέσεις (600 Δευτερόλεπτα).**

Παρατηρούμε ότι το LSTM δίκτυο κατάφερε με αρκετά μεγάλη ακρίβεια να προβλέψει κάθε φορά την επόμενη θέση του αεροπλάνου (και το γεωγραφικό πλάτος και το γεωγραφικό μήκος) ενώ όσο αυξάνει ο ο χρόνος που θέλουμε να προβλέψουμε στο μέλλον τόσο αυξάνει και αυτό το σφάλμα, γεγονός αναμενόμενο (εφόσον επίσης βλέπει και λιγότερα παραδείγματα το δίκτυο).



## **Βιβλιογραφία**

[1] Mina Moradi Kordmahalleh, Mohammad Gorji Sefidmazgi, and Abdollah Homaifar Department of Electrical Engineering, North Carolina A&T State University Greensboro, NC, USA, 27411 mmoradik@aggies.ncat.edu, mgorjis@gmail.com, [Homaifar@ncat.edu](mailto:Homaifar@ncat.edu). **A Sparse Recurrent Neural Network for Trajectory Prediction of Atlantic Hurricanes**, Denver, Colorado, USA — July 20 - 24, 2016, GECCO '16 Proceedings of the Genetic and Evolutionary Computation Conference 2016 Pages 957-964

[2] Jianming Lv, South China University of Technology,Guangzhou,China, Email: [jmlv@scut.edu.cn](mailto:jmlv@scut.edu.cn), Qing Li, City University of Hongkong, Email: qing.li@cityu.edu.hk, Xintong Wang, South China University of Technology,Guangzhou, China, Email: [csalexwang@mail.scut.edu.cn](mailto:csalexwang@mail.scut.edu.cn), **Modeling Trajectory as Image: Convolutional Neural Networks for Multi-scale Taxi Trajectory Prediction**, 2016 ArXiv

[3] Federico Bartoli, [federico.bartoli@unifi.it](mailto:federico.bartoli@unifi.it), Giuseppe Lisanti, [giuseppe.lisanti@unifi.it](mailto:giuseppe.lisanti@unifi.it), Lamberto Ballan, [lamberto.ballan@unipd.it](mailto:lamberto.ballan@unipd.it), Alberto Del Bimbo, [alberto.delbimbo@unifi.it](mailto:alberto.delbimbo@unifi.it) **Context-Aware Trajectory Prediction in Crowded Spaces**, Media Integration and Communication Center, University of Florence, Italy, BMVC 2017

[4] ByeoungDo Kim, Chang Mook Kang, Seung Hi Lee, Hyunmin Chae, Jaekyum Kim, Chung Choo Chung, and Jun Won Choi, Hanyang University, Seoul, Korea, Email: bdkim@spo.hanyang.ac.kr, kcm0728@hanyang.ac.kr, [shlee@ieee.org](mailto:shlee@ieee.org), fhmchae,jkming@spo.hanyang.ac.kr, cchung@hanyang.ac.kr and [junwchoi@hanyang.ac.kr](mailto:junwchoi@hanyang.ac.kr) **Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network**, arXiv:1704.07049v2 [cs.LG] 1 Sep 2017

[5] Tharindu Fernando , Simon Denman , Sridha Sridharan, Clinton Fookes, Image and Video Research Laboratory, SAIVT, Queensland University of Technology, Australia, **Soft + Hardwired Attention: An LSTM Framework for Human Trajectory Prediction and Abnormal Event Detection**, Published 2017 in ArXiv

[6] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, Silvio Savarese Stanford University, **Social LSTM: Human Trajectory Prediction in Crowded Spaces**, Computer Vision and Pattern Recognition (CVPR), 2016 IEEE, 27-30 June 2016, Las Vegas, NV, USA

[7] Michael Goldhammer, Konrad Doll, Ulrich Brunsmann, University of Applied Sciences Aschaffenburg, Aschaffenburg, Germany, Email:{michael.goldhammer, konrad.doll, ulrich.brunsmann}@h-ab.de, Andre Gensler, Bernhard Sick Intelligent Embedded Systems Lab University of Kassel, Kassel, Germany, Email: {gensler, [bsick](mailto:bsick@uni-kassel.de)}@uni-kassel.de, **Pedestrian's Trajectory Forecast in Public Traffic with Artificial Neural Networks**, Pattern Recognition (ICPR), 2014 22nd International Conference, 24-28 Aug. 2014, Stockholm, Sweden

[8] Alex Graves, Department of , Computer Science, University of Toronto, [graves@cs.toronto.edu](mailto:graves@cs.toronto.edu), **Generating Sequences With Recurrent Neural Networks**, arXiv:1308.0850, 4 Aug 2013

[9] Craig Schlenoff, Raj Madhavan, Stephen Balakirsky, Intelligent Systems Division, National Institute of Standards and Technology, 100 Bureau Drive, Stop 8230, Gaithersburg, MD 20899, Phone: 301-975-3456 F: 301-990-9688. [craig.schlenoff@nist.gov](mailto:craig.schlenoff@nist.gov), raj.madhavan@nist.gov, [stephen.balakirsky@nist.gov](mailto:stephen.balakirsky@nist.gov), **An Approach to Predicting the Location of Moving Objects During On-Road Navigation**, 18th International Joint Conference on Artificial Intelligence, August 9-15, 2003, Acapulco

[10] J.Venkata Subramanian, Dr.Abdul Karim Sadiq, Department of MCA, SRM University, Kattankulathur, Chennai, India, College of Applied Sciences, Sohar, Ministry of Higher



Education, Sultanate of Oman, **Mobile Location Prediction Methods – A Survey**, International Journal of Software and Web Sciences (IJSWS)

[11] International Journal of Software and Web Sciences (IJSWS), U.Sakthi and R.S.Bhuvaneshwaran, Department of Computer Science, Research Scholar, Anna University, Chennai 600 025, India Ramanujan Computing Center, Asst.Professor, Anna University, Chennai 600 025, India, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.1 , January 2009

[12] Mikołaj Morzy, Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 ,Poznan, Poland, [Mikolaj.Morzy@put.poznan.pl](mailto:Mikolaj.Morzy@put.poznan.pl), **Mining Frequent Trajectories of Moving Objects for Location Prediction**, MLDM '07 Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition Pages 667 – 680, July 18 - 20 2007, Leipzig, Germany

[13] Anna Monreale, Dept. Computer Science, University of Pisa, Italy, ISTI - CNR Pisa, Italy, [annam@di.unipi.it](mailto:annam@di.unipi.it), Roberto Trasarti, Dept. Computer Science, University of Pisa, Italy, ISTI - CNR Pisa, Italy, [roberto.trasarti@isti.cnr.it](mailto:roberto.trasarti@isti.cnr.it), Fabio Pinelli, ISTI - CNR Pisa, Italy, [fabio.pinelli@isti.cnr.it](mailto:fabio.pinelli@isti.cnr.it), Fosca Giannotti, ISTI - CNR Pisa, Italy, [fosca.giannotti@isti.cnr.it](mailto:fosca.giannotti@isti.cnr.it), **WhereNext: a Location Predictor on Trajectory Pattern Mining**, KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining Pages 637-646, June 28 - July 01 2009, Paris, France

[14] Zhixian Yan EPFL - Ecole Polytechnique Fédérale de Lausanne, Switzerland, [zhixian.yan@epfl.ch](mailto:zhixian.yan@epfl.ch), **Traj-ARIMA: A Spatial-Time Series Model for Network-Constrained Trajectory**, IWCTS '10 Proceedings of the Third International Workshop on Computational Transportation Science Pages 11-16, November 02 - 02 2010, San Jose, California

[15] Cynthia Sung, Dan Feldman, Daniela Rus, **Trajectory Clustering for Motion Prediction**, Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference, 7-12 Oct. 2012, Vilamoura, Portugal

[16] Joao Bartolo Gomes, Clifton Phua, Shonali Krishnaswamy, Institute for Infocomm Research (I2R), A\*STAR, Singapore, [fbartologjp](mailto:fbartologjp@sas.com), [spkrishnag@i2r.a-star.edu.sg](mailto:spkrishnag@i2r.a-star.edu.sg), SAS, Singapore [clifton.phua@sas.com](mailto:clifton.phua@sas.com), **Where will you go? Mobile Data Mining for Next Place Prediction**, DaWaK 2013 Proceedings of the 15th International Conference on Data Warehousing and Knowledge Discovery - Volume 8057 Pages 146-158, August 26 - 29 2013, Prague, Czech Republic

[17] Gokhan Yavas, Dimitrios Katsaros , Ozgur Ulusoy, Yannis Manolopoulos, Department of Computer Engineering, Bilkent University, Bilkent, Ankara 06533, Turkey, Department of Informatics, Aristotle University, Thessaloniki, Greece, **A data mining approach for location prediction in mobile environments**, Journal: Data & Knowledge Engineering archive Volume 54 Issue 2 Pages 121 - 146, August 2005

[18] <http://cs231n.github.io/neural-networks-1/>

[19] <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>

[20] <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[20] Dynamic Time Warping Algorithm (Slides From: Elena Tsiporkova)