



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

---

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

Διαγράμματα ελέγχου, μοντέλα χρονοσειρών και  
αναδρομικά νευρωνικά δίκτυα στην αξιοπιστία συστημάτων

---

Φανή Πετροπούλου  
Αριθμός μητρώου : 09116028

Επιβλέπων Καθηγητής:  
Χρήστος Κουκουβίνος, Καθηγητής ΕΜΠ

Τριμελής επιτροπή:  
Χρήστος Κουκουβίνος, Καθηγητής ΕΜΠ  
Χρυσή Καρώνη, Καθηγήτρια ΕΜΠ  
Πέτρος Στεφανέας, Επίκουρος Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2021

---

©(2021) Εθνικό Μετσόβιο Πολυτεχνείο.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



NATIONAL TECHNICAL UNIVERSITY OF ATHENS  
SCHOOL OF MATHEMATICAL AND PHYSICAL SCIENCES

---

## THESIS

---

**Control charts, Time series models and Recurent neural  
networks in reliability of analytical systems**

---

Fani Petropoulou  
Registration Number : 09116028

**Supervisor Professor:**  
Christos Koukouvinos, NTUA Professor

**Three Member Commitee:**  
Christos Koukouvinos, NTUA Professor  
Chrysseis Caroni, NTUA Professor  
Petros Stefaneas, NTUA Lecturer

Athens, July 2021

---

©(2021) National Technical University of Athens.  
All rights Reserved.

It is prohibited to copy, store and distribute this work, in whole or in part, for commercial purposes. Reprinting, storing and distributing for non-profit, educational or research purposes is permitted, as long as there is reference to the source and retain the present message. Questions regarding the use of this work for profit purposes should be addressed to the author. The views and conclusions contained in this document are those of the author and should not be construed as representing the official positions of the National Technical University of Athens.

## Περίληψη

Η κλασική μέθοδος για τον έλεγχο της αξιοπιστίας ενός συστήματος είναι η επανάληψη της ανάλυσης ενός δείγματος αναφοράς σε τακτά χρονικά διαστήματα και η παρακολούθηση του μέσω του στατιστικού ελέγχου διεργασιών (Statistical Process Control - SPC). Οι συνήθεις τεχνικές του SPC που προϋποθέτουν την ανεξαρτησία των δεδομένων, έχουν αποδειχθεί ανεπαρκείς για συσχετισμένα δεδομένα όπως είναι οι χρονοσειρές. Στο σημείο αυτό, τα νευρωνικά δίκτυα εισάγονται ως μία καινοτόμα και σύγχρονη λύση για την μοντελοποίηση και αξιολόγηση τέτοιου είδους διεργασιών.

Η παρούσα εργασία, εξετάζει την ικανότητα ενός αναδρομικού νευρωνικού δικτύου να μοντελοποιεί χρονοσειρές με σκοπό τον υπολογισμό υπολοίπων (σφάλματα πρόβλεψης). Εν συνεχεία, τα υπόλοιπα, τα οποία ικανοποιούν τις προϋποθέσεις κανονικότητας και ανεξαρτησίας, εξετάζονται από τις κλασικές μεθόδους του SPC. Η μεθοδολογία αυτή συγκρίνεται με κάποια άλλα διαγράμματα που έχουν προταθεί στη βιβλιογραφία.

Η οργάνωση της διπλωματικής εργασίας έχει ως ακολούθως. Στο πρώτο κεφάλαιο, γίνεται μία εισαγωγή στις τεχνικές του SPC που έχουν ήδη προταθεί για την παρακολούθηση διεργασιών συσχετισμένων δεδομένων. Στο δεύτερο κεφάλαιο, αναλύεται η έννοια της χρονοσειράς καθώς και κάποιες βασικές μέθοδοι μοντελοποίησης της. Στο τρίτο κεφάλαιο, περιγράφονται τα κλασικά διαγράμματα ελέγχου ανεξάρτητων παρατηρήσεων και πως αυτά τροποποιήθηκαν ώστε να εφαρμοστούν σε εξαρτημένα δεδομένα. Επίσης στο κεφάλαιο αυτό γίνεται αναφορά στο μέσο μήκος ροής (Average Run Length) το οποίο χρησιμοποιείται ως μέτρο σύγκρισης των μεθόδων που θα εξετάσουμε στη συνέχεια. Στο τέταρτο κεφάλαιο, αναλύεται ο τρόπος λειτουργίας των αναδρομικών νευρωνικών δικτύων. Στο πέμπτο και τελευταίο κεφάλαιο, εκτελούνται οι συγκρίσεις των μεθόδων που περιγράψαμε στα προηγούμενα κεφάλαια μέσω προσομοιώσεων Monte Carlo και σχολιάζονται τα τελικά αποτελέσματα.

**Λέξεις Κλειδιά** - Στατιστικός Έλεγχος Διεργασιών, Διαγράμματα ελέγχου, Χρονοσειρές, Αναδρομικά Νευρωνικά Δίκτυα

---

## Abstract

The classic method for checking the reliability of a system is to repeat the analysis of a reference sample at regular intervals and monitor it through the Statistical Process Control (SPC). The usual SPC techniques that require data independence have been shown to be insufficient for correlated data such as time series. At this point, neural networks are being introduced as an innovative and modern solution for modeling and evaluating such processes.

This thesis examines the ability of a recurrent neural network to model time series in order to calculate residuals (prediction errors). Then, the residuals, which satisfy the conditions of regularity and independence, are imported to the usual control charts of SPC. This methodology is compared with some other charts that have been proposed in the literature.

This thesis has been organized as follows. In the first chapter, an introduction to the SPC techniques that have already been proposed for monitoring correlated data processes is carried out. In the second chapter, the concept of time series is developed as well as some time series models. The third chapter describes some classical control charts for independent data and how they have been modified in order to be applied to correlated observations. Also in this chapter, the Average Run Length (ARL) is introduced, which is used as a performance metric to the methods that we will examine in the last chapter. The fourth chapter delineates the structure and function of recurrent neural networks. In the fifth and last chapter, the comparisons of the methods described in the previous chapters are performed through Monte Carlo simulations and final results are contrasted and commented.

**Keywords** - Statistical Process Control, Control Charts, Time series, Recurrent neural networks

---



## Ευχαριστίες

Οφείλω ιδιαίτερες ευχαριστίες, αρχικά στον καθηγητή και επιβλέποντα της διπλωματικής μου εργασίας, τον Κ. Χρήστο Κουκουβίνο για την εμπιστοσύνη που επέδειξε και την στήριξη που παρείχε καθώς και στην υποψήφια διδάκτορα του Εθνικού Μετσόβιου Πολυτεχνείου, Αγγελική Λάππα, η οποία ήταν πάντοτε πρόθυμη να βοηθήσει.

Σπουδαία συμβολή για την επιτυχημένη εκπόνηση της εργασίας αυτής είχε ο Σπύρος Παρασκευάς, ή όπως πολύ του αρέσει να τον αποκαλώ Κύριος Σπύρος Παρασκευάς, καθώς και ο αγαπητός φίλος και συμφοιτητής Θεμιστοκλής Νικόλης.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένεια και τους φίλους μου για την κατανόηση και την υποστήριξη τους με κάθε τρόπο όλα αυτά τα χρόνια των σπουδών μου.

*“Sapere aude”*

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>1</b>
<b>2</b>	<b>Χρονοσειρές και Μοντέλα Χρονοσειρών</b>	<b>3</b>
2.1	Χρονοσειρές . . . . .	3
2.2	Μοντέλα Χρονοσειρών . . . . .	7
2.2.1	Θόρυβος . . . . .	7
2.2.1.1	Θόρυβος IID . . . . .	7
2.2.1.2	White noise . . . . .	7
2.2.2	Μοντέλο ARIMA $(p, d, q)$ . . . . .	9
2.2.2.1	Τελεστής $B$ . . . . .	11
2.2.2.2	Συνθήκη στασιμότητας . . . . .	12
2.2.2.3	Αντιστρεψιμότητα . . . . .	12
2.2.3	Seasonal ARIMA $(p, d, q) \times (P, D, Q)_m$ . . . . .	13
2.2.4	Μοντέλο ETS . . . . .	14
2.2.4.1	Simple Exponential Smoothing . . . . .	14
2.2.4.2	Η μέθοδος γραμμικής τάσης του Holt . . . . .	16
2.2.4.3	Η μέθοδος εποχικότητας του Holt και Winter . . . . .	17
<b>3</b>	<b>Διαγράμματα ελέγχου</b>	<b>19</b>
3.1	Μέσο μήκος ροής - ARL . . . . .	19
3.2	Διαγράμματα του SPC για ανεξάρτητες παρατηρήσεις . . . . .	20
3.2.1	Διάγραμμα EWMA . . . . .	21
3.2.2	Διάγραμμα CUSUM . . . . .	22
3.3	Απόδοση των διαγραμμάτων σε εξαρτημένες παρατηρήσεις . . . . .	25
3.4	Διαγράμματα του SPC για εξαρτημένες παρατηρήσεις . . . . .	26
3.4.1	Διαγράμματα υπολοίπων . . . . .	26
3.4.2	Τροποποιημένο διάγραμμα EWMA . . . . .	28
3.4.3	Τροποποιημένο διάγραμμα CUSUM . . . . .	31
<b>4</b>	<b>Τεχνητά νευρωνικά δίκτυα - ANN</b>	<b>33</b>
4.1	Λειτουργία νευρωνικών δικτύων . . . . .	35
4.1.1	Forward propagation . . . . .	43
4.1.2	Backward propagation . . . . .	43
4.1.3	Το πρόβλημα της υπερπροσαρμογής . . . . .	53
4.2	Αναδρομικά νευρωνικά δίκτυα . . . . .	54
4.2.1	ERNN . . . . .	54
4.2.2	LSTM . . . . .	56
4.2.3	GRU . . . . .	59

4.2.4	Forward propagation . . . . .	61
4.2.5	Backward propagation . . . . .	61
4.2.6	Bi-Directional RNN . . . . .	62
4.2.7	Deep RNN . . . . .	64
<b>5</b>	<b>Περιγραφή πειράματος και παρουσίαση αποτελεσμάτων</b>	<b>65</b>
5.1	Σύγκριση για διαδικασία $AR(1)$ . . . . .	69
5.1.1	Σχολιασμός αποτελεσμάτων . . . . .	74
5.2	Σύγκριση για διαδικασία $ARIMA(1, 0, 1)$ . . . . .	78
5.2.1	Σχολιασμός αποτελεσμάτων . . . . .	80
5.3	Σύγκριση για διαδικασία $SARIMA(1, 0, 0)(1, 0, 0)_4$ . . . . .	82
5.3.1	Σχολιασμός αποτελεσμάτων . . . . .	85
5.4	Σύγκριση για διαδικασία $ETS(A,N,N)$ . . . . .	88
5.4.1	Σχολιασμός αποτελεσμάτων . . . . .	90
5.5	Σύνοψη αποτελεσμάτων . . . . .	91
5.6	Επίλογος και μελλοντική έρευνα . . . . .	94
	<b>Βιβλιογραφία</b>	<b>95</b>

## Λίστα Σχημάτων

2.1	Σταθερότητα μέσης τιμής και διασποράς . . . . .	4
2.2	Διάσπαση της χρονοσειράς σε συνιστώσες . . . . .	6
2.3	Μορφή χρονοσειράς white noise . . . . .	7
3.1	Απόδοση των διαγραμμάτων σε εξαρτημένες παρατηρήσεις . . . . .	25
4.1	Τα είδη στρωμάτων του νευρωνικού δικτύου . . . . .	33
4.2	Εποπτική περιγραφή της λειτουργίας του νευρώνα . . . . .	35
4.3	Λογιστική σιγμοειδής συνάρτηση ενεργοποίησης . . . . .	36
4.4	Παράγωγος της λογιστικής σιγμοειδούς συνάρτησης ενεργοποίησης . . . . .	37
4.5	Συνάρτηση ενεργοποίησης $\tanh(x)$ . . . . .	39
4.6	Συνάρτηση ενεργοποίησης ReLu . . . . .	40
4.7	Συνάρτηση ενεργοποίησης Leaky ReLu . . . . .	41
4.8	Νευρωνικό δίκτυο με 2 κρυφά στρώματα . . . . .	42
4.9	Ορθή επιλογή του ρυθμού εκμάθησης . . . . .	45
4.10	Οπτική απεικόνιση της μεθόδου Gradient Descent . . . . .	46
4.11	Οπτική απεικόνιση της μεθόδου Mini Batch Gradient Descent . . . . .	47
4.12	Κελί Elman RNN . . . . .	55
4.13	Κελί LSTM . . . . .	58
4.14	Κελί GRU . . . . .	60
4.15	Forward Propagation . . . . .	61
4.16	Backward Propagation through time . . . . .	62
4.17	Δομή ενός BRNN . . . . .	63
4.18	Δομή ενός Deep RNN . . . . .	64
5.1	$ARL_1$ για συσχέτιση $\phi_1 = 0.8$ στα διάφορα διαγράμματα . . . . .	74
5.2	$ARL_1$ για συσχέτιση $\phi_1 = 0.2$ στα διάφορα διαγράμματα . . . . .	75
5.3	$ARL_1$ για συσχέτιση $\phi_1 = -0.2$ στα διάφορα διαγράμματα . . . . .	76
5.4	$ARL_1$ για συσχέτιση $\phi_1 = -0.8$ στα διάφορα διαγράμματα . . . . .	77
5.5	$ARL_1$ για συσχέτιση $\phi_1 = -0.5$ και $\theta = -0.5$ στα διάφορα διαγράμματα . . . . .	80
5.6	$ARL_1$ για συσχέτιση $\phi_1 = 0.5$ και $\theta = 0.5$ στα διάφορα διαγράμματα . . . . .	81
5.7	$ARL_1$ για συσχέτιση $\phi_1 = -0.5$ και $\Phi_1 = -0.4$ στα διάφορα διαγράμματα . . . . .	85
5.8	$ARL_1$ για συσχέτιση $\phi_1 = -0.5$ και $\Phi_1 = 0.4$ στα διάφορα διαγράμματα . . . . .	86
5.9	$ARL_1$ για συσχέτιση $\phi_1 = 0.5$ και $\Phi_1 = -0.4$ στα διάφορα διαγράμματα . . . . .	86
5.10	$ARL_1$ για συσχέτιση $\phi_1 = 0.5$ και $\Phi_1 = 0.4$ στα διάφορα διαγράμματα . . . . .	87
5.11	$ARL_1$ για το μοντέλο ETS(A,N,N) με $\alpha = 0.5$ στα διάφορα διαγράμματα . . . . .	90

## Λίστα Πινάκων

5.1	Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη $ARL_0 \approx 370$ . . . . .	72
5.2	Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη $ARL_0 \approx 370$ . . . . .	72
5.3	Σύγκριση διαγραμμάτων με βάση $ARL_1$ για μία AR(1) διαδικασία, όταν $ARL_0 = 370$ . . . . .	73
5.4	Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη $ARL_0$ σε διαδικασία ARIMA(1, 0, 1) . . . . .	79
5.5	Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη $ARL_0$ σε διαδικασία ARIMA(1, 0, 1) . . . . .	79
5.6	Σύγκριση διαγραμμάτων με βάση $ARL_1$ για μία ARIMA(1, 0, 1) διαδικασία, όταν $ARL_0 \approx 370$ . . . . .	79
5.7	Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη $ARL_0$ σε διαδικασία SARIMA(1, 0, 0)(1, 0, 0) <sub>4</sub> . . . . .	83
5.8	Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη $ARL_0$ σε διαδικασία SARIMA(1, 0, 0)(1, 0, 0) <sub>4</sub> . . . . .	83
5.9	Σύγκριση διαγραμμάτων με βάση $ARL_1$ για μία SARIMA(1, 0, 0)(1, 0, 0) <sub>4</sub> διαδικασία, όταν $ARL_0 \approx 370$ . . . . .	84
5.10	Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη $ARL_0 \approx 370$ σε μία διαδικασία ETS(A,N,N) . . . . .	89
5.11	Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη $ARL_0 \approx 370$ σε μία διαδικασία ETS(A,N,N) . . . . .	89
5.12	Σύγκριση διαγραμμάτων με βάση $ARL_1$ για μία ETS(A,N,N) διαδικασία, όταν $ARL_0 \approx 370$ . . . . .	89

# 1 Εισαγωγή

Τα τελευταία χρόνια, ο στατιστικός έλεγχος διεργασιών (Statistical Process Control - SPC) ή αλλιώς στατιστικός έλεγχος ποιότητας (Statistical Quality Control - SQC) για αυτοσυσχετισμένες διαδικασίες έχει λάβει μεγάλη προσοχή στην ακαδημαϊκή έρευνα και τη βιομηχανία. Αυτό οφείλεται εν μέρει στις βελτιώσεις στη μέτρηση και τη συλλογή δεδομένων που επιτρέπουν τη δειγματοληψία διεργασιών σε υψηλότερη συχνότητα και, ως εκ τούτου, την ανάδειξη της αυτοσυσχέτισης των δεδομένων.

Τα παραδοσιακά διαγράμματα ελέγχου (Shewhart, Cumulative Sum (CUSUM), Exponentially Weighted Moving Average (EWMA) ) έχουν χρησιμοποιηθεί ευρέως ως αποτελεσματικά εργαλεία στον SPC. Ωστόσο, σχεδόν όλες αυτές οι προτεινόμενες μέθοδοι υποθέτουν ότι τα δεδομένα της διαδικασίας είναι ανεξάρτητα και ισόνομα όταν η διαδικασία είναι υπό έλεγχο. Με άλλα λόγια, τα δεδομένα μιας διεργασίας θεωρείται ότι δεν σχετίζονται μεταξύ τους. Ωστόσο δεν είναι εφικτή η εφαρμογή τέτοιων μεθόδων σε περιπτώσεις διεργασιών που παρουσιάζουν αυτοσυσχέτιση (π.χ. λειτουργικοί δείκτες επίδοσης εγκαταστάσεων, οχημάτων, συσκευών). Έχουν γίνει αρκετές προσπάθειες για την επέκταση των παραδοσιακών τεχνικών SPC για την αντιμετώπιση των συσχετισμένων δεδομένων. Ωστόσο, αυτές οι επεκτάσεις θέτουν αρκετούς και σοβαρούς περιορισμούς .

Γενικά, όπως αναφέρεται στους Pacella και Semeraro (2007), Chen και Yu (2019) και Loredó (2002), μια ευρέως χρησιμοποιούμενη προσέγγιση για την παρακολούθηση της στατιστικής διαδικασίας στην περίπτωση των συσχετισμένων δεδομένων είναι το διάγραμμα υπολοίπων (residual chart). Αυτό το διάγραμμα προϋποθέτει τον προσδιορισμό μιας επαρκούς μεθόδου μοντελοποίησης των παρατηρήσεων της υπόψη διεργασίας, ώστε τα υπόλοιπα (residuals) που θα προκύψουν να είναι κατάλληλα για θεώρηση. Παραδείγματα τέτοιων μεθόδων μοντελοποίησης είναι τα: Autoregressive models (AR), Moving Average models (MA) και Autoregressive Integrated Moving Average models (ARIMA), τα οποία αποτελούν μέρος του APC (Advanced process control). Παρ'όλα αυτά όμως, έχουν γίνει προσπάθειες για διεύρυνση της βάσης προσαρμογής του μοντέλου στο deep learning και συγκεκριμένα με την χρήση RNN (Recurrent Neural Networks). Διάφορες μελέτες και συγκρίσεις της παραπάνω μεθόδου (RNN-residual chart) με εκείνες του APC (Rius et al. (1998) ) δείχνουν ότι στις περισσότερες περιπτώσεις, η μέθοδος των residual chart-RNN ανταποκρίνεται καλύτερα.

Οι Wu και Yu (2010) πρότειναν μία μέθοδος βασισμένη σε νευρωνικό δίκτυο (NN-based) για την ανίχνευση μεταβολών στη μέση τιμή και τη διακύμανση σε συσχετισμένες διαδικασίες, με το όνομα DPSOEN (Discrete Particle Swarm Optimization algorithm based selective Network Ensemble). Το μοντέλο είναι σε θέση να παρακολουθεί τις μετατοπίσεις των μέσων και των διακυμάνσεων και να ταξινομεί τους τύπους αλλαγών. Πιο συγκεκριμένα, στο μοντέλο αυτό χρησιμοποιείται ένας αλγόριθμος βελτιστοποίησης DPSO (Discrete Intelligent

Particle Swarm Optimization) για την απόκτηση ενός βέλτιστου υποσυνόλου δικτύων. Ο Guh (2008) πρότεινε μία μέθοδος που βασίζεται πάλι σε νευρωνικό δίκτυο (NN-based) που μπορεί να αναγνωρίζει αποτελεσματικά τα μοτίβα στα διαγράμματα ελέγχου σε πραγματικό χρόνο για διαδικασίες με διάφορα επίπεδα αυτοσυσχέτισης. Αργότερα οι Purintrapiban και Corley (2012), πρότειναν άλλη μία NN-based μέθοδο για την αναγνώριση κυκλικών μοτίβων στο διάγραμμα ελέγχου μιας αυτοσυσχετισμένης διαδικασίας, καθώς η κυκλική συμπεριφορά αποτελεί ένδειξη ότι μια κατάσταση βρίσκεται εκτός ελέγχου. Τέλος, οι Qiu et al. (2020) πρότειναν ένα νέο διάγραμμα το οποίο δεν απαιτεί κανένα παραμετρικό μοντέλο χρονοσειρών και/ή μια παραμετρική κατανομή της διεργασίας, παρά μόνο την υπόθεση ότι η παρατήρηση της διαδικασίας σε ένα δεδομένο χρονικό σημείο σχετίζεται με κοντινές παρατηρήσεις και είναι ανεξάρτητη από παρατηρήσεις που βρίσκονται σε απομακρυσμένους χρόνους.

Η μοντελοποίηση των δεδομένων είναι σημαντικό να γίνεται έτσι, ώστε να επιτυγχάνουμε βέλτιστη προβλεπτική ικανότητα. Για τον λόγο αυτό, εισερχόμαστε σε ένα άλλο πεδίο που ονομάζεται βαθιά μάθηση (deep learning), επίσης γνωστή και ως βαθιά νευρωνικά δίκτυα (DNN - Deep Neural Networks). Αυτή είναι μια νέα μέθοδος εκμάθησης με πολλαπλά κρυμμένα επίπεδα αναπαράστασης. Πιο συγκεκριμένα, το επαναλαμβανόμενο νευρωνικό δίκτυο (Recurrent Neural Network-RNN) είναι ένα DNN που είναι ικανό να μελετά δεδομένα χρονοσειρών και να πραγματοποιεί προβλέψεις για το μοντέλο ώστε έπειτα να είμαστε σε θέση να μελετήσουμε τα residual charts, για την ανίχνευση μετατοπίσεων. Οι Chen και Yu (2019) πρότειναν μια μέθοδο βασισμένη σε RNN για τον εντοπισμό μετατοπίσεων των μέσων αυτοσυσχετισμένων διεργασιών παραγωγής. Οι Mulder et al. (2015) παρουσίασαν μία βιβλιογραφική έρευνα η οποία καλύπτει τις εφαρμογές των RNN στη στατιστική μοντελοποίηση της φυσικής γλώσσας, σημειώνοντας κάποιες πρόσφατες εξελίξεις τους με τις οποίες επιτυγχάνονται μικρότεροι χρόνοι εκπαίδευσης, ενώ επιπλέον κατέστη δυνατός ο περιορισμός των λέξεων που είναι απαραίτητες να ληφθούν υπόψη για τη διαμόρφωση επαρκούς νοηματικού πλαισίου. Γενικότερα, στη βιβλιογραφία, το RNN έχει εφαρμοστεί επιτυχώς σε μοντέλα πολύπλοκων χρονοεξαρτώμενων συστημάτων σε συνεχείς χημικές διεργασίες, οικονομικά δεδομένα, αναγνώριση λέξεων κ.λπ. Εμείς ενδιαφερόμαστε να μελετήσουμε την αξιοπιστία διαδικασιών αυτοσυσχετισμένων παρατηρήσεων, και συγκεκριμένα χρονοσειρών, μέσω κατάλληλων διαγραμμάτων που βασίζονται στην μοντελοποίηση των δεδομένων με χρήση RNN. Έπειτα θα είμαστε σε θέση να τα συγκρίνουμε με διάφορα διαγράμματα και μεθοδολογίες που έχουν ήδη αναδειχθεί στην διεθνή βιβλιογραφία.

## 2 Χρονοσειρές και Μοντέλα Χρονοσειρών

### 2.1 Χρονοσειρές

Μια χρονοσειρά ορίζεται ως μία συλλογή παρατηρήσεων  $x_t$ ,  $t \in T_0$  όπου ο δείκτης  $t$  δηλώνει την χρονική στιγμή που λάβαμε την αντίστοιχη παρατήρηση και  $T_0$  είναι το σύνολο της χρονικής διάρκειας.

Γενικά, η χρονοσειρά ακολουθεί κάποιο συγκεκριμένο μοντέλο που περιγράφει την από κοινού κατανομή των τ.μ.  $x_t$ . Έτσι, η μαθηματική έκφραση που περιγράφει τη δομή μιας χρονοσειράς ονομάζεται στοχαστική διαδικασία  $\{X_t : t \in T\}$  με  $T_0 \subset T$ . Η ακολουθία των παρατηρήσεων της σειράς είναι στην πραγματικότητα ένα στιγμιότυπο της στοχαστικής διαδικασίας που την παρήγαγε.

Μια χρονοσειρά μπορεί να είναι συνεχής ή διακριτή. Για παράδειγμα, οι μετρήσεις της θερμοκρασίας, η ροή ενός ποταμού, η συγκέντρωση μιας χημικής διαδικασίας κ.τ.λ., μπορούν να καταγραφούν ως συνεχείς χρονοσειρές. Από την άλλη πλευρά, ο πληθυσμός μιας συγκεκριμένης πόλης, η παραγωγή μιας εταιρείας, οι συναλλαγματικές ισοτιμίες μεταξύ δύο διαφορετικών νομισμάτων, μπορεί να αντιπροσωπεύουν διακριτές χρονοσειρές. Συνήθως σε μια διακριτή χρονοσειρά οι διαδοχικές παρατηρήσεις καταγράφονται σε ισόποσα χρονικά διαστήματα όπως ωριαία, ημερήσια, εβδομαδιαία, μηνιαία ή ετήσια. Γενικότερα, μια συνεχής χρονοσειρά μπορεί εύκολα να μετατραπεί σε διακριτή με τη συγχώνευση δεδομένων σε ένα καθορισμένο χρονικό διάστημα.

### Στασιμότητα

Ένα σημαντικό είδος χρονοσειρών είναι οι στάσιμες.

Μία στοχαστική διαδικασία  $\{X_t\}$  (ή απλά χρονοσειρά) λέγεται (ασθενώς) στάσιμη αν:

$$E(X_t) = \mu \quad \forall t \in \mathbb{Z}$$

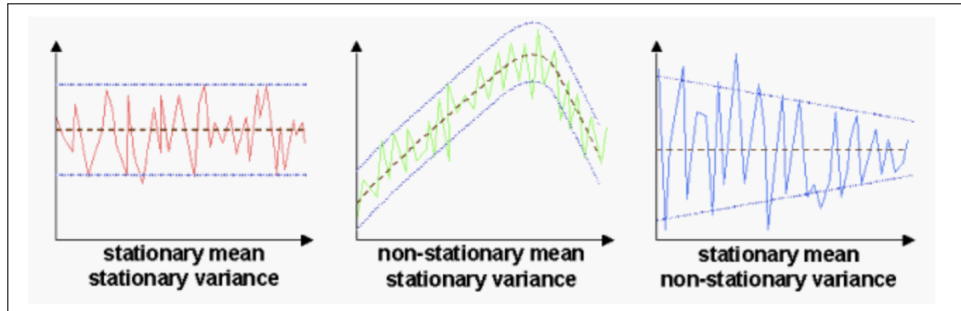
$$Cov(X_t, X_{t+h}) = \gamma_h \quad \forall t, t+h \in \mathbb{Z}$$

Με απλά λόγια, μία διεργασία καλείται στάσιμη αν ο μέσος και η συνδιακύμανση (για κάθε  $h$ ) δεν εξαρτώνται από τον χρόνο. Από την τελευταία σχέση φυσικά συνεπάγεται ότι  $Var(X_t) = Cov(X_t, X_t) = \gamma_0 \rightarrow$  σταθερό. Το  $h$  ονομάζεται lag (χρονική υστέρηση). Δηλαδή το  $k$ -οστό lag είναι η χρονική περίοδος που συνέβη “ $k$ ” χρονικές στιγμές (ημέρες, μήνες, χρόνια) πριν από την χρονική στιγμή  $t$ .

Για παράδειγμα:  $Lag_1(X_2) = X_1$  και  $Lag_4(X_9) = X_5$ .



Σημαντικά χαρακτηριστικά μιας στάσιμης χρονοσειράς είναι ο μέσος  $\mu$ , η συνάρτηση  $\gamma_h$  (autocovariance function-ACVF) και η συνάρτηση αυτοσυσχέτισης  $\rho_h = \frac{\gamma_h}{\gamma_0}$  (autocorrelation function -ACF).



Σχήμα 2.1: Σταθερότητα μέσης τιμής και διασποράς

## Θεώρημα

Βασικές ιδιότητες της συνάρτησης  $\gamma_h$  είναι οι εξής:

1.  $\gamma_0 \geq 0$
2.  $|\gamma_h| \leq \gamma_0 \quad \forall h$
3.  $\gamma_h = \gamma_{(-h)} \quad \forall h$
4.  $\gamma_h$  είναι θετικά ορισμένη.

Μία συνάρτηση  $K$  θεωρείται θετικά ορισμένη αν και μόνον αν

$$\sum_{i,j=1}^n a_i K(i-j) a_j \geq 0 \quad \forall n, \alpha = (\alpha_1, \dots, \alpha_n)^T \in \mathbb{R}^n.$$

Απόδειξη:

$$1. \gamma_0 = Cov(X_t, X_t) = Var(X_t) \geq 0 \quad \forall t.$$

2. Σύμφωνα με την ανισότητα Cauchy-Schwartz:

$$|\gamma_h| = |Cov(X_t, X_{t+h})| \leq \sqrt{Var(X_t)} \sqrt{Var(X_{t+h})} = \gamma_0$$

3. Εύκολα παρατηρούμε ότι:

$$\gamma_h = Cov(X_t, X_{t+h}) = Cov(X_{t+h}, X_t) = \gamma_{(-h)}$$

4.

$$0 \leq \text{Var}(\alpha^\top X_n) = \alpha^\top \Gamma_n \alpha = \sum_{i,j=1}^n \alpha_i \gamma_{(i-j)} \alpha_j$$

όπου  $X_n = (X_n, \dots, X_1)^\top$  και

$$\Gamma_n = \text{Var}(X_n) = \begin{pmatrix} \text{Cov}(X_n, X_n) & \text{Cov}(X_n, X_{n-1}) & \dots & \text{Cov}(X_n, X_2) & \text{Cov}(X_n, X_1) \\ \text{Cov}(X_{n-1}, X_n) & \text{Cov}(X_{n-1}, X_{n-1}) & \dots & \text{Cov}(X_{n-1}, X_2) & \text{Cov}(X_{n-1}, X_1) \\ \vdots & \ddots & & & \\ \text{Cov}(X_2, X_n) & \text{Cov}(X_2, X_{n-1}) & \dots & \text{Cov}(X_2, X_2) & \text{Cov}(X_2, X_1) \\ \text{Cov}(X_1, X_n) & \text{Cov}(X_1, X_{n-1}) & \dots & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_1) \end{pmatrix}$$

$$= \begin{pmatrix} \gamma_0 & \gamma_1 & \dots & \gamma_{n-2} & \gamma_{n-1} \\ \gamma_1 & \gamma_0 & \dots & \gamma_{n-3} & \gamma_{n-2} \\ \vdots & \ddots & & & \\ \gamma_{n-2} & \gamma_{n-3} & \dots & \gamma_0 & \gamma_1 \\ \gamma_{n-1} & \gamma_{n-2} & \dots & \gamma_1 & \gamma_0 \end{pmatrix}$$

Η συνάρτηση αυτοσυσχέτισης  $\rho(\cdot)$  έχει ακριβώς τις ίδιες ιδιότητες και επιπλέον  $\rho(0) = 1$ .

### Συνιστώσες μιας χρονοσειράς

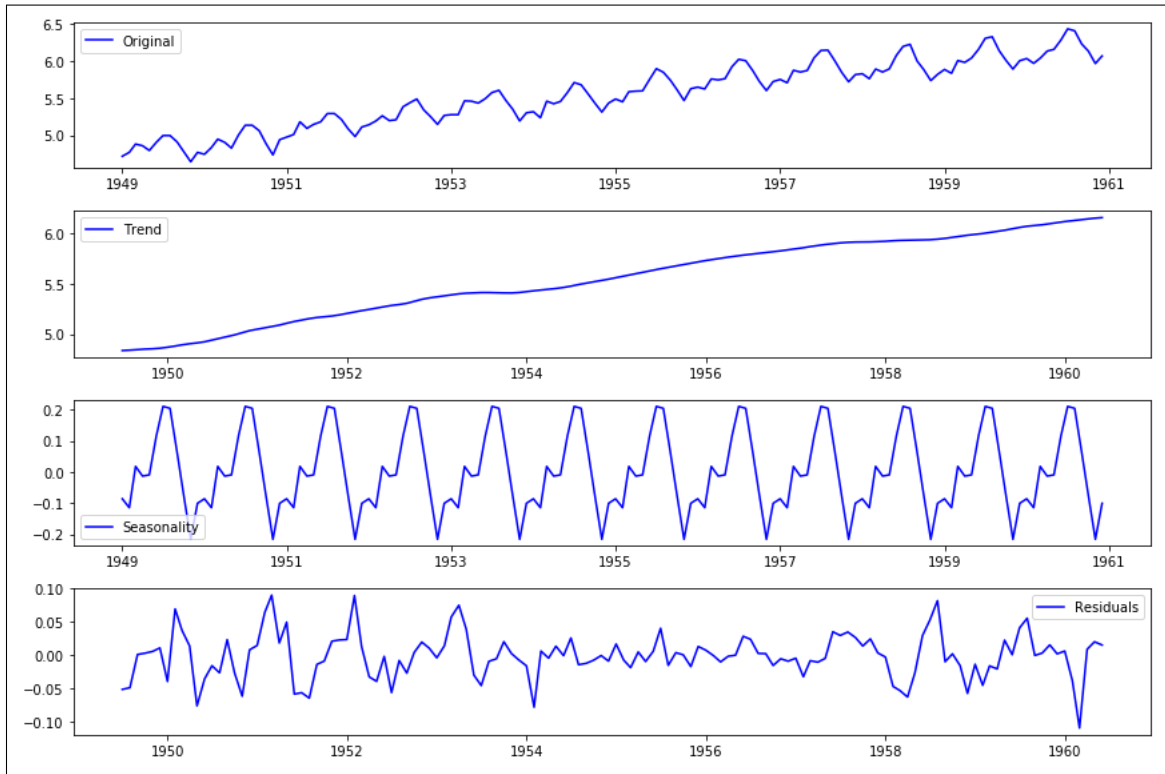
Μια χρονολογική σειρά επηρεάζεται από τρία κύρια στοιχεία, τα οποία μπορούν να διαχωριστούν από τα παρατηρούμενα δεδομένα. Αυτά τα στοιχεία είναι: Τάσεις (trends), κύκλοι (cycles), εποχικότητα (seasonality).

**Τάση (trend):** Η γενική τάση μιας χρονοσειράς να αυξάνεται, να μειώνεται ή να σταματά για μεγάλο χρονικό διάστημα.

**Εποχικότητα (seasonality):** Οι εποχιακές διακυμάνσεις σε μια χρονοσειρά είναι διακυμάνσεις εντός ενός έτους κατά τη διάρκεια της σεζόν. Βασικοί παράγοντες που προκαλούν εποχιακές διακυμάνσεις είναι: το κλίμα και οι καιρικές συνθήκες, τα έθιμα, οι παραδοσιακές συνήθειες κ.λπ.

**Κύκλοι (cycles):** Η κυκλική διακύμανση σε μια χρονοσειρά περιγράφει τις μεσοπρόθεσμες αλλαγές στα δεδομένα που προκαλούνται από διάφορες περιστάσεις, οι οποίες επαναλαμβάνονται σε κύκλους. Διαφέρουν από την εποχικότητα καθώς η διάρκεια ενός κύκλου εκτείνεται για περισσότερο χρόνο, συνήθως δύο ή περισσότερα χρόνια και δεν υπάρχει επανάληψη σε συγκεκριμένο χρονικό διάστημα.

Αν αφαιρέσουμε από την χρονοσειρά τις παραπάνω συνιστώσες, στο τέλος μένει το **υπόλοιπο (remainder component)**.



Σχήμα 2.2: Διάσπαση της χρονοσειράς σε συνιστώσες

## 2.2 Μοντέλα Χρονοσειρών

### 2.2.1 Θόρυβος

#### 2.2.1.1 Θόρυβος IID

Το πιο απλό μοντέλο χρονοσειρών είναι αυτό που δεν έχει τάση ή εποχιακό στοιχείο και οι παρατηρήσεις  $X_t$  είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές με μη-δενικό μέσο όρο. Μια τέτοια ακολουθία τυχαίας μεταβλητής  $\{X_t\}$  ονομάζεται θόρυβος IID (Ανεξάρτητα και Ισόνομα Κατανομημένος - Independent and Identically Distributed).

Για κάθε  $t$  και  $x_1, \dots, x_t$ :

$$P(X_1 \leq x_1, \dots, X_t \leq x_t) = \prod_t P(X_t \leq x_t) = \prod_t F(x_t)$$

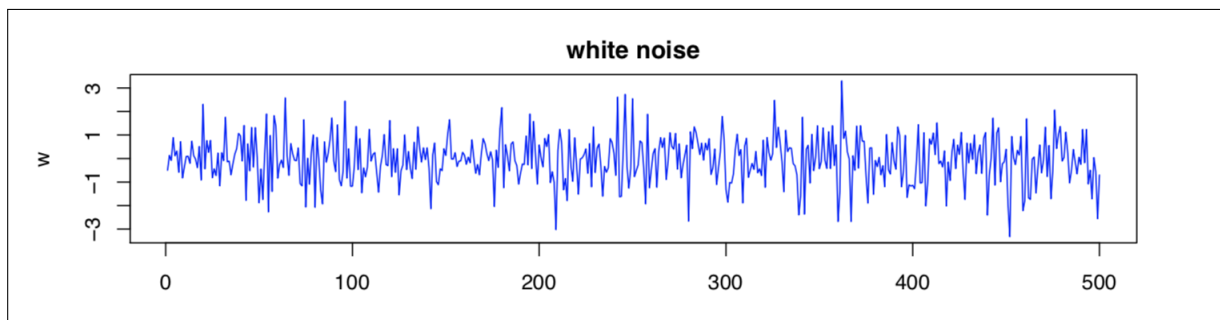
όπου  $F(\cdot)$  είναι η συνάρτηση κατανομής των  $X_t$ . Επίσης ισχύει  $E(X_t) = 0 \forall t$ .

Συμβολισμός:  $X_t \sim IID(0, \sigma^2)$ .

#### 2.2.1.2 White noise

Μία ακολουθία  $\{X_t\}$  ορίζεται ως white noise (λευκός θόρυβος) αν:  
 $Cov(X_{t_1}, X_{t_2}) = 0 \forall t_1, t_2$  με  $E[X_t] = 0$  και  $Var[X_t] = \sigma^2$ .

Με άλλα λόγια οι παρατηρήσεις μας είναι ασυσχέτιστες με μέση τιμή 0 και σταθερή διασπορά  $\sigma^2$ .



Σχήμα 2.3: Μορφή χρονοσειράς white noise

Συμβολισμός:  $X_t \sim WN(0, \sigma^2)$ .

**Σημείωση:** Κάθε  $IID(0, \sigma^2)$  ακολουθία είναι  $WN(0, \sigma^2)$  χωρίς να ισχύει απαραίτητα το αντίστροφο.

Για παράδειγμα, έστω  $X_t \sim WN(0, \sigma^2)$ . Επίσης, έστω  $X_t = Z_t$ , όταν  $t$  περιττό και  $X_t = \sqrt{3}Z_{t-1}^2 - \frac{2}{\sqrt{3}}$ , όταν  $t$  άρτιο. Ορίζουμε ως  $\{Z_t, t = 1, 3, \dots\} \sim IID$  με συνάρτηση μάζας πιθανότητας  $f_Z(-1) = \frac{1}{3}$ ,  $f_Z(0) = \frac{1}{3}$  και  $f_Z(1) = \frac{1}{3}$ . Είναι εύκολο να δείξουμε ότι:

**Για  $t$  περιττό:**

$$E(X_t) = E(Z_t) = \sum_{x \in \{-1, 0, 1\}} x f_Z(x) = (-1) \cdot \frac{1}{3} + 0 \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} = 0$$

$$E(X_t^2) = E(Z_t^2) = \sum_{x^2 \in \{-1, 0, 1\}} x^2 f_Z(x) = (-1)^2 \cdot \frac{1}{3} + 0^2 \cdot \frac{1}{3} + 1^2 \cdot \frac{1}{3} = \frac{2}{3}$$

**Για  $t$  άρτιο:**

$$E(X_t) = E\left(\sqrt{3}Z_{t-1}^2 - \frac{2}{\sqrt{3}}\right) = \sqrt{3}E(Z_{t-1}^2) - \frac{2}{\sqrt{3}} = \sqrt{3} \sum_{x \in \{-1, 0, 1\}} x^2 f_Z(x) - \frac{2}{\sqrt{3}} =$$

$$= \sqrt{3} \frac{2}{3} - \frac{2}{\sqrt{3}} = 0$$

$$E(X_t^2) = E\left[\left(\sqrt{3}Z_{t-1}^2 - \frac{2}{\sqrt{3}}\right)^2\right] = 3E(Z_{t-1}^4) - 4E(Z_{t-1}^2) + \frac{4}{3} = \frac{6}{3} - \frac{8}{3} + \frac{4}{3} = \frac{2}{3}$$

Άρα μπορούμε να συμπεράνουμε ότι:

$$E(t) = 0, \forall t \text{ και } Var(X_t) = E(X_t^2) - E(X_t)^2 = \frac{2}{3}, \forall t.$$

Επίσης,  $Cov(X_{t_1}, X_{t_2}) = 0$ ,  $\forall t_1, t_2$  καθώς:

$$Cov(Z_t, \sqrt{3}Z_{t-1}^2 - \frac{2}{\sqrt{3}}) = \sqrt{3}Cov(Z_t, Z_t^2) = 0. \text{ Δηλαδή } X_t \sim WN(0, \frac{2}{3}).$$

Παρ' όλα αυτά δεν ισχύει ότι  $X_t \sim IDD$  καθώς το  $Z_{2k}$  καθορίζεται πλήρως από το  $Z_{2k-1}$ :

$$Z_{2k-1} = 0 \Leftrightarrow Z_{2k} = \frac{-2}{\sqrt{3}}$$

$$Z_{2k-1} = \pm 1 \Leftrightarrow Z_{2k} = \sqrt{3} - \frac{2}{\sqrt{3}}$$

### 2.2.2 Μοντέλο ARIMA $(p, d, q)$

#### Autoregressive model (p) - AR (p)

Υποθέτουμε ότι  $\{X_t\}$  στάσιμη χρονοσειρά.

Σε ένα μοντέλο AR (p) θεωρούμε ότι η μελλοντική τιμή μιας μεταβλητής είναι ένας γραμμικός συνδυασμός προηγούμενων παρατηρήσεων.

Συγκεκριμένα το AR (p) μοντέλο μπορεί να γραφεί ως εξής:

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \epsilon_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \epsilon_t$$

όπου  $\epsilon_t \sim WN(0, \sigma^2)$ ,  $\phi_i$  η παράμετρος του μοντέλου και  $c$  σταθερά.

Ο όρος  $p$  είναι γνωστός ως η τάξη του AR μοντέλου.

#### Moving Average model (q) - MA (q)

Υποθέτουμε ότι  $\{X_t\}$  στάσιμη χρονοσειρά.

Ένα μοντέλο MA (q) χρησιμοποιεί προηγούμενα σφάλματα ως επεξηγηματικές μεταβλητές. Δηλαδή:

$$X_t = \mu + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \epsilon_t = \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t$$

όπου  $\mu$  είναι ο μέσος όρος των παρατηρήσεων της χρονοσειράς,  $\epsilon_t \sim WN(0, \sigma^2)$  και  $\theta_j$  η παράμετρος του μοντέλου.

Ο όρος  $q$  είναι γνωστός ως η τάξη του MA μοντέλου.

### Autoregressive Moving Average model (p,q) - ARMA (p,q)

Υποθέτουμε ότι  $\{X_t\}$  στάσιμη χρονοσειρά.

Τα μοντέλα Autoregressive (AR) και κινητού μέσου όρου (MA) μπορούν να συνδυαστούν αποτελεσματικά και να σχηματίσουν μια γενική και χρήσιμη κατηγορία μοντελοποίησης χρονοσειρών, γνωστά ως **μοντέλα ARMA (p,q)**:

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \epsilon_t$$

Προφανώς αν  $p = 0$ , το παραπάνω μοντέλο είναι MA (q), ενώ αν  $q = 0$  είναι AR (p).

### Autoregressive Integrated Moving Average (p,d,q) - ARIMA (p,d,q)

Όλα τα παραπάνω μοντέλα, είχαν την προϋπόθεση η χρονοσειρά που εφαρμόζονται να είναι στάσιμη. Στην περίπτωση όμως που αυτή δεν είναι, θα πρέπει να μετασχηματιστεί. Εδώ εισέρχεται η χρησιμότητα του όρου Integrated ή αλλιώς differencing. Δηλαδή ένας τρόπος ώστε να μετατρέψω την χρονοσειρά σε στάσιμη είναι να υπολογίσω τη διαφορά ανάμεσα σε διαδοχικές παρατηρήσεις. Η διαφορά αυτή μπορεί να συμβάλλει στη σταθεροποίηση του μέσου όρου της χρονοσειράς με την κατάργηση (ή μείωση) της τάσης και της εποχικότητας της. Σε κάποιες περιπτώσεις μάλιστα δεν αρκεί να πάρουμε μία διαφορά, αλλά περισσότερες. Έτσι έχουμε:

$$\text{Διαφορά } 1^{\text{ης}} \text{ τάξης (d=1)} : X'_t = X_t - X_{t-1}$$

$$\begin{aligned} \text{Διαφορά } 2^{\text{ης}} \text{ τάξης (d=2)} : X''_t &= X'_t - X'_{t-1} = (X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) \\ &= X_t - 2X_{t-1} + X_{t-2} \end{aligned}$$

.

.

κ.ο.κ.

Το μοντέλο ARIMA (p,d,q) μπορεί να γραφεί ως:

$$X'_t = c + \sum_{i=1}^p \phi_i X'_{t-i} + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \epsilon_t$$

όπου  $X'_t$  συμβολίζω τη χρονοσειρά που έχει υποστεί κάποια διαφορά τάξης  $d$ ,  $\epsilon_t \sim WN(0, \sigma^2)$ ,  $\phi_i$  και  $\theta_j$  οι παράμετροι του μοντέλου και  $c$  σταθερά.

Είναι προφανές ότι:

- Ένα μοντέλο AR (p) ταυτίζεται με ARIMA(p,0,0)
- Ένα μοντέλο MA (q) ταυτίζεται με ARIMA(0,0,q)
- Ένα μοντέλο ARMA (p,q) ταυτίζεται με ARIMA(p,0,q)

### 2.2.2.1 Τελεστής $B$ - Backshift Operator

Συνήθως, για ευκολία, χρησιμοποιούμε τον τελεστή  $B$  ως εξής:

$$BX_t = X_{t-1}$$

Έτσι έχουμε:

- AR(p):  $\epsilon_t + c = \phi(B)X_t$
- MA(q):  $X_t = \mu + \theta(B)\epsilon_t$
- ARMA(p,q):  $\phi(B)X_t = \theta(B)\epsilon_t + c$
- ARIMA(p,d,q):  $\phi(B)(1 - B)^d X_t = \theta(B)\epsilon_t + c$

όπου τα  $\phi(B) = 1 - \sum_{i=1}^p \phi_i B^i$  και  $\theta(B) = 1 + \sum_{j=1}^q \theta_j B^j$  ονομάζονται χαρακτηριστικά πολυώνυμα.



### 2.2.2.2 Συνθήκη στασιμότητας

Για να έχουμε στάσιμη χρονοσειρά θα πρέπει οι παράμετροι των μοντέλων να ικανοποιούν κάποιες εξισώσεις. Συγκεκριμένα, για το ARIMA (p,d,q) θα πρέπει οι ρίζες του χαρακτηριστικού πολυωνύμου  $\phi(B) = 0$  να βρίσκονται έξω από τον μοναδιαίο κύκλο.

Για παράδειγμα, για το AR(1):

$$\phi(B) = 0 \Leftrightarrow 1 - \phi_1 B = 0 \Leftrightarrow B = \frac{1}{\phi_1}$$

Για να χαρακτηρίζει μια στάσιμη χρονοσειρά θα πρέπει  $|B| > 1 \Leftrightarrow |\phi_1| < 1$ , ενώ για το AR(2), όμοια, πρέπει  $|\phi_2| < 1$ ,  $\phi_1 + \phi_2 < 1$  και  $\phi_2 - \phi_1 < 1$ .

Το MA(q) ικανοποιεί πάντα την συνθήκη στασιμότητας.

### 2.2.2.3 Αντιστρεψιμότητα

Μία σημαντική ιδιότητα των AR(p) μοντέλων είναι η αντιστρεψιμότητα τους. Δηλαδή κάθε AR(p) μπορεί να γραφεί ως ένα MA( $\infty$ ). Αντίθετα όμως, το MA(q) και κατ'επέκταση το ARMA(p,q) δεν είναι πάντοτε αντιστρέψιμα. Για να ισχύει αυτό θα πρέπει οι ρίζες της χαρακτηριστικής εξίσωσης  $\theta(B) = 0$  να βρίσκονται εκτός του μοναδιαίου κύκλου. Για παράδειγμα, το MA(1) είναι αντιστρέψιμο (δηλαδή μπορεί να γραφεί ως AR( $\infty$ )) αν  $|\theta_1| < 1$ , ενώ το MA(2) αν  $|\theta_2| < 1$ ,  $\theta_1 + \theta_2 > -1$  και  $\theta_1 - \theta_2 < 1$ . Έτσι, το ARIMA (p,d,q) θα είναι αντιστρέψιμο αν το MA (q) μέρος της είναι αντιστρέψιμο.

### 2.2.3 Seasonal Autoregressive Integrated Moving Average $(p, d, q) \times (P, D, Q)_m$ - SARIMA $(p, d, q) \times (P, D, Q)_m$

Το μοντέλο ARIMA  $(p, d, q)$  που περιγράφηκε παραπάνω αφορά σε δεδομένα που δεν παρουσιάζουν την συνιστώσα της εποχικότητας. Οι Box και Jenkins (1970) εισήγαγαν το μοντέλο SARIMA, το οποίο είναι ικανό να επεξεργάζεται χρονοσειρές με εποχικότητα (seasonality). Οι όροι  $(p, d, q)$  αφορούν το μη-εποχιακό μέρος του μοντέλου, ακριβώς όπως τους ορίσαμε στο μοντέλο ARIMA, ενώ οι υπόλοιποι  $(P, D, Q)_m$  αφορούν το εποχιακό μέρος, όπου  $m$  το σύνολο των παρατηρήσεων ανά έτος. Οι όροι  $(P, D, Q)$  είναι εντελώς αντίστοιχοι με εκείνους του απλού ARIMA, με τη διαφορά ότι αφορούν τις παρατηρήσεις ίδιας εποχής.

- Αν  $P = 1$ , δηλαδή ένα μοντέλο πρώτης τάξης seasonal autoregressive, τότε θα χρησιμοποιηθεί η τιμή της μεταβλητής  $X_{t-m}$  για την πρόβλεψη της  $X_t$ , ενώ αν  $P = 2$  θα χρησιμοποιηθούν οι  $X_{t-m}$  και  $X_{t-2m}$ .
- Αν  $D = 1$ , τότε θα έχουμε την διαφορά μιας παρατήρησης με την αντίστοιχη του προηγούμενου έτους, δηλαδή  $X'_t = X_t - X_{t-m}$ . Αν  $D = 2$ , τότε  $X''_t = X'_t - X'_{t-m} = X_t - X_{t-m} - (X_{t-m} - X_{t-2m}) = X_t - 2X_{t-m} + X_{t-2m}$ , κ.ο.κ.
- Αν  $Q = 1$ , δηλαδή ένα μοντέλο πρώτης τάξης seasonal moving average, θα χρησιμοποιηθούν για την πρόβλεψη τα σφάλματα  $\epsilon_{t-m}$ . Αν  $Q = 2$ , χρησιμοποιούνται τα  $\epsilon_{t-m}$  και  $\epsilon_{t-2m}$ , κ.ο.κ.

Για μηνιαία δεδομένα ορίζουμε  $m = 12$ , ενώ για τριμηνιαία  $m = 4$ .

Με τη βοήθεια του τελεστή  $B$ , το μοντέλο SARIMA περιγράφεται από την εξίσωση:

$$\boxed{\Phi(B)\phi(B)(1-B)^d(1-B^m)^D X_t = \Theta(B)\theta(B)\epsilon_t + c}$$

, όπου  $\Phi(B) = 1 - \sum_{i=1}^P \Phi_i B^{im}$  και  $\Theta(B) = 1 + \sum_{j=1}^Q \Theta_j B^{jm}$  τα χαρακτηριστικά πολυώνυμα του εποχιακού μέρους του μοντέλου και  $\phi(B), \theta(B)$  τα χαρακτηριστικά πολυώνυμα ARIMA όπως τα ορίσαμε προηγουμένως.

### 2.2.4 Μοντέλο ETS (Exponential smoothing)

Τα μοντέλα ETS (Error, Trend, Seasonality) είναι μια οικογένεια μοντέλων χρονοσειρών βασισμένο σε τρεις συνιστώσες, το σφάλμα (Error), την τάση (Trend) και την εποχικότητα (Seasonality).

#### 2.2.4.1 SES (Simple Exponential Smoothing)

Το απλούστερο μοντέλο της οικογένειας αυτής είναι το Simple Exponential Smoothing (SES), το οποίο είναι κατάλληλο όταν τα δεδομένα μας δεν εμφανίζουν κάποια τάση (trend) ή εποχικότητα (seasonality). Δίνεται από τον εξής αναδρομικό τύπο:

$$S_t = \alpha X_t + (1 - \alpha)S_{t-1} = S_{t-1} + \alpha(X_t - S_{t-1})$$

όπου  $0 \leq \alpha \leq 1$  η παράμετρος εξομάλυνσης,  $X_t$  οι παρατηρήσεις της χρονοσειράς μας και  $t = 1, 2, \dots, T$  η χρονική στιγμή που λαμβάνουμε την παρατήρηση.

Αν συμβολίσουμε με  $S_0 = l_0$  την πρώτη τιμή που μας δίνει το μοντέλο, τότε έχουμε:

$$\begin{aligned} S_1 &= \alpha X_1 + (1 - \alpha)l_0 \\ S_2 &= \alpha X_2 + (1 - \alpha)S_1 \\ &= \alpha X_2 + (1 - \alpha)(\alpha X_1 + (1 - \alpha)l_0) \\ &= \alpha X_2 + \alpha(1 - \alpha)X_1 + (1 - \alpha)^2 l_0 \\ S_3 &= \alpha X_3 + (1 - \alpha)S_2 \\ &= \alpha X_3 + (1 - \alpha)(\alpha X_2 + \alpha(1 - \alpha)X_1 + (1 - \alpha)^2 l_0) \\ &= \alpha X_3 + \alpha(1 - \alpha)X_2 + \alpha(1 - \alpha)^2 X_1 + (1 - \alpha)^3 l_0 \\ &\vdots \\ S_t &= \sum_{j=0}^{t-1} \alpha(1 - \alpha)^j X_{t-j} + (1 - \alpha)^t l_0 \end{aligned}$$

Η παραπάνω μορφή αποτελεί ουσιαστικά έναν σταθμισμένο μέσο (weighted average form) ανάμεσα στην πιο πρόσφατη και τις προηγούμενες παρατηρήσεις. Όσο μεγαλύτερη είναι η τιμή της παραμέτρου  $\alpha$  τόσο λιγότερη είναι και η εξομάλυνση, καθώς και το βάρος που δίνεται στις προγενέστερες παρατηρήσεις.

Μία άλλη μορφή με την οποία μπορούμε να παρουσιάσουμε το SES μοντέλο είναι η μορφή συνιστωσών (Component form).

Εξίσωση πρόβλεψης (Forecast equation):

$$\hat{X}_{t+h} = l_t$$

Εξίσωση εξομάλυνσης (Smoothing equation):

$$l_t = \alpha X_t + (1 - \alpha)l_{t-1}$$

όπου  $l_t$  είναι το επίπεδο (ή η εξομαλυμένη τιμή) της παρατήρησης μας την χρονική στιγμή  $t$  και  $h$  είναι το διάστημα πρόβλεψης. Αν  $h = 1$  προφανώς θα έχουμε τις προσαρμοσμένες τιμές (fitted values).

Η μορφή αυτή δεν είναι ιδιαίτερα χρήσιμη, αλλά θα είναι πιο εύχρηστη παρακάτω που θα προσθέσουμε κι άλλες συνιστώσες όπως η τάση και η εποχικότητα.

Το σφάλμα του μοντέλου μπορεί να βρίσκεται σε δύο μορφές:

### 1. Προσθετικά σφάλματα (Additive errors)

Αν αναλύσουμε την εξίσωση εξομάλυνσης:

$$l_t = \alpha X_t + (1 - \alpha)l_{t-1} = l_{t-1} + \alpha(X_t - l_{t-1}) = l_{t-1} + \alpha\epsilon_t$$

Τότε τα υπόλοιπα υπολογίζονται ως:

$$\epsilon_t = X_t - l_{t-1} = X_t - \hat{X}_{t|t-1}$$

### 2. Πολλαπλασιαστικά σφάλματα (Multiplicative errors)

Στην περίπτωση αυτή τα υπόλοιπα υπολογίζονται ως τα σχετικά σφάλματα:

$$\epsilon_t = \frac{X_t - \hat{X}_{t|t-1}}{\hat{X}_{t|t-1}}$$

Αντικαθιστώντας  $\hat{X}_{t|t-1} = l_{t-1}$  θα έχουμε τις νέες εξισώσεις:

$$X_t = l_{t-1} + l_{t-1}\epsilon_t = l_{t-1}(1 + \epsilon_t)$$

$$\begin{aligned} l_t &= l_{t-1} + \alpha(X_t - l_{t-1}) \\ &= l_{t-1} + \alpha(\epsilon_t l_{t-1}) \\ &= l_{t-1}(1 + \alpha\epsilon_t) \end{aligned}$$

#### 2.2.4.2 Η μέθοδος γραμμικής τάσης του Holt (Holt's linear trend method)

Ο Holt (1957) επέκτεινε το μοντέλο SES ώστε να μπορεί να περιγράψει δεδομένα που παρουσιάζουν κάποια τάση. Παρακάτω φαίνονται οι εξισώσεις που περιγράφουν το επεκτεταμένο αυτό μοντέλο.

Εξίσωση πρόβλεψης (Forecast equation):

$$\hat{X}_{t+h} = l_t + hb_t$$

Εξίσωση εξομάλυνσης (Smoothing equation):

$$l_t = \alpha X_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

Εξίσωση τάσης (Trend equation):

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$$

, όπου  $l_t$  είναι το επίπεδο (ή η εξομαλυμένη τιμή) της παρατήρησης μας την χρονική στιγμή  $t$ ,  $h$  είναι το διάστημα πρόβλεψης,  $b_t$  η εκτίμηση της τάσης (κλίσης) τη χρονική στιγμή  $t$ ,  $0 \leq \alpha \leq 1$  η παράμετρος εξομάλυνσης για το επίπεδο  $l_t$  και  $0 \leq \beta^* \leq 1$  η παράμετρος εξομάλυνσης για την τάση.

Όμοια με το SES, η μεταβλητή  $l_t$  είναι ένας σταθμισμένος μέσος της παρατήρησης  $X_t$  και του όρου  $l_{t-1} + b_{t-1}$ , ενώ η μεταβλητή  $b_t$  αποτελεί έναν σταθμισμένο μέσο του όρου  $l_t - l_{t-1}$  και της προηγούμενης εκτίμησης  $b_{t-1}$ .

**Παρατήρηση:** Η συνάρτηση-εξίσωση πρόβλεψης  $\hat{X}_{t+h}$  δεν είναι πλέον σταθερή αλλά παρουσιάζει γραμμική σχέση με την τιμή της σταθεράς  $h$ .

### 2.2.4.3 Η μέθοδος εποχικότητας του Holt και Winter (Holt-Winter's seasonal method)

Οι Holt (1957) και Winter (1960) επέκτειναν το μοντέλο του Holt έτσι ώστε να συμπεριλάβουν σε αυτό τη συνιστώσα της εποχικότητας. Υπάρχουν δύο παραλλαγές σε αυτήν τη μέθοδο που διαφέρουν ως προς τη φύση του εποχιακού στοιχείου. Η προσθετική μέθοδος (additive method), η οποία προτιμάται όταν οι εποχιακές διακυμάνσεις είναι περίπου σταθερές κατά τη διάρκεια της εξέλιξης της χρονοσειράς και η πολλαπλασιαστική μέθοδος (multiplicative method), η οποία προτιμάται όταν η εποχικότητα αλλάζει ανάλογα με το επίπεδο της χρονοσειράς.

#### Προσθετική μέθοδος

Με τη μέθοδο αυτή, το εποχιακό στοιχείο εκφράζεται σε απόλυτους όρους στην κλίμακα των παρατηρήσεων και στην εξίσωση εξομάλυνσης η σειρά προσαρμόζεται αφαιρώντας το εποχιακό στοιχείο. Θα έχουμε:

Εξίσωση πρόβλεψης (Forecast equation):

$$\hat{X}_{t+h} = l_t + hb_t + s_{t+h-m(k+1)}$$

Εξίσωση εξομάλυνσης (Smoothing equation):

$$l_t = \alpha(X_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$

Εξίσωση τάσης (Trend equation):

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$$

Εξίσωση εποχικότητας (Seasonality equation):

$$s_t = \gamma(X_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$$

, όπου  $m$  είναι η συχνότητα της εποχικότητας,  $0 \leq \gamma \leq 1$  η παράμετρος εξομάλυνσης για την εποχικότητα  $s_t$ ,  $h$  το διάστημα πρόβλεψης και  $k$  είναι το ακέραιο μέρος της ποσότητας  $\frac{h-1}{m}$  και διασφαλίζει ότι οι εκτιμήσεις των εποχιακών δεικτών που χρησιμοποιούνται για την πρόβλεψη προέρχονται από το τελευταίο έτος του δείγματος.

Παρατήρηση: Η συνιστώσα της εποχικότητας  $s_t$  θα αθροίζει περίπου στο μηδέν ανά έτος.

## Πολλαπλασιαστική μέθοδος

Με την πολλαπλασιαστική μέθοδο, το εποχιακό στοιχείο εκφράζεται σε σχετικούς όρους (ποσοστά) και η σειρά προσαρμόζεται εποχιακά διαιρώντας με την εποχικότητα. Θα έχουμε:

Εξίσωση πρόβλεψης (Forecast equation):

$$\hat{X}_{t+h} = (l_t + hb_t)s_{t+h-m(k+1)}$$

Εξίσωση εξομάλυνσης (Smoothing equation):

$$l_t = \alpha \frac{X_t}{s_{t-m}} + (1 - \alpha)(l_{t-1} + b_{t-1})$$

Εξίσωση τάσης (Trend equation):

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$$

Εξίσωση εποχικότητας (Seasonality equation):

$$s_t = \gamma \frac{X_t}{l_{t-1} - b_{t-1}} + (1 - \gamma)s_{t-m}$$

Παρατήρηση: Η συνιστώσα της εποχικότητας  $s_t$  θα αθροίζει περίπου στο  $m$  ανά έτος.

Φυσικά τα ETS μοντέλα δεν περιορίζονται στα παραπάνω, αλλά μπορούν να προκύψουν ως συνδυασμοί προσθετικότητας και πολλαπλασιαστικότητας ανάμεσα στις διάφορες συνιστώσες (σφάλμα, τάση, εποχικότητα).

### 3 Διαγράμματα ελέγχου

Βασικά εργαλεία του στατιστικού ελέγχου ποιότητας είναι τα διαγράμματα ελέγχου που στόχος τους είναι να ανιχνεύουν μετατοπίσεις κατά τη διάρκεια της εξέλιξης μιας διεργασίας. Τα βασικότερα από αυτά είναι το Shewhart, το EWMA (Exponentially Weighted Moving Average) και το CUSUM (Cumulative Sum). Στην περίπτωση των αυτοσυσχετισμένων δεδομένων όπως είναι οι χρονοσειρές, θα δούμε ότι κρίνεται απαραίτητο τα διαγράμματα αυτά να τροποποιηθούν ώστε να ελαχιστοποιήσουμε τις εσφαλμένες ενδείξεις.

#### 3.1 Μέσο μήκος ροής - ARL (Average Run Length)

Ένα βασικό μέτρο υπολογισμού της απόδοσης των διαγραμμάτων είναι το μέσο μήκος ροής (ARL).

Το ARL (Average Run Length-Μέσο μήκος ροής) ορίζεται ως ο αναμενόμενος αριθμός σημείων που πρέπει να απεικονισθούν στο διάγραμμα, μέχρις ότου να έχουμε ένδειξη εκτός ελέγχου.

Το ARL χωρίζεται σε **εντός ελέγχου- $ARL_0$**  και **εκτός ελέγχου- $ARL_1$** .

#### **$ARL_0$**

Το  $ARL_0$  είναι το εντός ελέγχου μήκος ροής, δηλαδή εκφράζει κάθε πόσα δείγματα θα έχουμε ένδειξη (εσφαλμένη) ότι η διεργασία βρίσκεται εκτός ελέγχου.

Δίνεται από:

$$ARL_0 = \frac{1}{\alpha}$$

όπου  $\alpha = P[X \notin (LCL, UCL) | X \sim P(c)]$ : η πιθανότητα να έχουμε ένδειξη ότι η διεργασία είναι εκτός ελέγχου, ενώ στην πραγματικότητα είναι εντός.



**ARL<sub>1</sub>**

Το  $ARL_1$  είναι το εκτός ελέγχου μέσο μήκος ροής, δηλαδή δηλώνει τον αναμενόμενο αριθμό δειγμάτων που πρέπει να ληφθούν ώστε να εντοπιστεί η μετατόπιση που προέκυψε στο μέσο της διεργασίας.

Δίνεται από:

$$ARL_1 = \frac{1}{1 - \beta}$$

όπου  $\beta = P[X \in (LCL, UCL) | X \sim P(c^*)]$ : η πιθανότητα να έχουμε ένδειξη (εσφαλμένη) ότι η διεργασία είναι εντός ελέγχου, ενώ στην πραγματικότητα είναι εκτός.

Γενικά θέλουμε μεγάλο  $ARL_0$ , ώστε σε μία εντός ελέγχου διεργασία να χρειαστεί μεγάλος αριθμός δειγμάτων μέχρι την (εσφαλμένη) ένδειξη ότι η διεργασία βρίσκεται εκτός ελέγχου. Αντιθέτως, επιδιώκουμε μικρό  $ARL_1$ , ώστε να γίνεται γρήγορα αντιληπτή μία εκτός ελέγχου διεργασία.

**3.2 Διαγράμματα του SPC για ανεξάρτητες παρατηρήσεις**

Θα παρουσιάσουμε σύντομα τα διαγράμματα ελέγχου Shewhart, CUSUM και EWMA στην περίπτωση ανεξάρτητων παρατηρήσεων  $X_t$ . Ουσιαστικά, το Shewhart είναι μία υποπερίπτωση του EWMA και χρησιμοποιείται κυρίως για ανίχνευση μεγάλων μετατοπίσεων. Θα θεωρήσουμε το μοντέλο:

$$X_t = Y_t + \sqrt{\gamma_0} \alpha I_{m, m+1, \dots}(t), \quad t \geq 1 \quad (1)$$

όπου  $I$  είναι μία δείκτρια συνάρτηση (1 αν  $t \geq m$  και 0 διαφορετικά),  $\gamma_0 = Var(Y_t) = \sigma^2$ ,  $\alpha$  το μέγεθος της μετατόπισης και  $\{Y_t\}$  μία άλλη στοχαστική διαδικασία-στόχος.

Η τιμή στόχος θα είναι  $\mu = E(Y_t)$ . Επίσης, υποθέτουμε ότι τα  $\mu$  και  $\gamma_0 = Var(Y_t)$  είναι γνωστά. Η διεργασία  $\{X_t\}$  θα είναι εντός ελέγχου αν  $\alpha = 0$ . Τότε θα ισχύει  $E(X_t) = E(Y_t) = \mu$ ,  $\forall t \geq 1$ . Διαφορετικά, η διεργασία είναι εκτός ελέγχου (για  $t \geq m$ ).

### 3.2.1 Διάγραμμα EWMA (Exponentially Weighted Moving Average)

Ο S. W. Roberts (1959) εισήγαγε , ένα διάγραμμα ελέγχου γνωστό ως EWMA. Αυτό χαρακτηρίζεται από την εξής ακολουθία:

$$Z_t = (1 - \lambda)Z_{t-1} + \lambda X_t, \quad t \geq 1 \quad (2)$$

Θέτουμε  $Z_0 = z_0 = \mu = E(Y_t)$ .

Η παράμετρος  $\lambda \in (0, 1]$  είναι γνωστή ως συντελεστής βαρύτητας (weighted factor) και καθορίζει το βαθμό κατά τον οποίο «παλιότερα» δεδομένα εισάγονται στον υπολογισμό του EWMA στατιστικού. Όσο πιο κοντά στη μονάδα είναι η τιμή της παραμέτρου αυτής, τόσο μικρότερο βάρος δίνεται στα προγενέστερα δεδομένα.

Αν  $\lambda = 1$ , τότε έχουμε το γνωστό διάγραμμα Shewhart. Αυτό σημαίνει ότι μόνο η πιο πρόσφατη παρατήρηση επηρεάζει το στατιστικό.

Αναπτύσσοντας περαιτέρω την σχέση (2), καταλήγουμε στον εξής τύπο:

$$Z_t = z_0(1 - \lambda)^t + \lambda \sum_{i=0}^{t-1} (1 - \lambda)^{t-i} X_i$$

Φαίνεται ότι όσο αυξάνεται το  $t$  τόσο μειώνεται το βάρος που δίνεται στα προγενέστερα δεδομένα.

Θεωρώντας ότι τα  $X_t$  είναι ανεξάρτητες τ.μ. με διασπορά  $\sigma^2$ , τότε η διασπορά των  $Z_t$  θα είναι:

$$\sigma_t^2 = \sigma^2 \left( \frac{\lambda}{2 - \lambda} \right) (1 - (1 - \lambda)^{2t})$$

Τα όρια ελέγχου, άνω - UCL (Upper control limit) και κάτω - LCL (Lower control limit), καθώς και η κεντρική γραμμή - CL (Center line) του διαγράμματος υπολογίζονται ως:

$$UCL = \mu_0 + L\sigma \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}$$

$$CL = \mu_0$$

$$LCL = \mu_0 - L\sigma \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}$$

όπου  $L$  το εύρος των ορίων και  $\mu_0$  η τιμή στόχος.

Είναι σημαντικό να παρατηρήσουμε ότι  $1 - (1 - \lambda)^{2t} \rightarrow 1$  για  $t \rightarrow \infty$ . Αυτό έχει ως συνέπεια, από ένα σημείο και μετά να εξομαλύνονται τα όρια (δηλαδή χωρίς τον όρο  $(1 - (1 - \lambda)^{2t})$  ο οποίος τα καθιστούσε μεταβλητά).

Αν βρεθεί  $Z_t > UCL$  ή  $Z_t < LCL$ , τότε η διεργασία καθίσταται εκτός ελέγχου.

Το πρόβλημα στο EWMA διάγραμμα, έγκειται στη σωστή επιλογή των παραμέτρων  $\lambda$  και  $L$ , έτσι ώστε να έχουμε την επιθυμητή ARL απόδοση. Επειδή ο προσδιορισμός των παραμέτρων  $\lambda$  και  $L$  για το EWMA καθίσταται περίπλοκος, έχουν γίνει αρκετές θεωρητικές έρευνες για να βρεθούν κάποιες τεχνικές υπολογισμού τους. Οι Lucas και Saccucci (1990) για γνωστό μέγεθος αλλαγής (μετατόπιση) στο μέσο της διεργασίας, κατάφεραν να βρουν μέσω προσομοιώσεων τη βέλτιστη τιμή του  $\lambda$  και  $L$ , κρατώντας σταθερό το  $ARL_0$  και ελαχιστοποιώντας το  $ARL_1$ .

### 3.2.2 Διάγραμμα CUSUM (Cumulative Sum)

Τα διαγράμματα CUSUM εισήχθησαν από τον Page (1954) και συνδέονται με το SPRT (Sequential Probability ratio test).

Έστω  $Y_t \sim N(\mu, \sigma^2)$  για  $t \geq 1$ . Για  $m = 1$ , η (1) συνεπάγεται ότι  $X_t \sim N(\mu + \alpha\sigma, \sigma^2)$ . Αρχικά, θεωρούμε τον εξής έλεγχο υποθέσεων:

$$H_0 : \alpha = 0$$

$$H_1 : \alpha = \alpha_0 \text{ για γνωστό } \alpha_0 > 0.$$

Υπενθυμίζουμε ότι αν  $\alpha = 0$ , η διεργασία θεωρείται εντός ελέγχου.

Κατά το SPRT η δειγματοληψία σταματάει τη στιγμή  $t$  αν:

$$\sum_{i=1}^t T_i \notin [A, B]$$

$$\text{με } T_i = \alpha_0 \frac{X_i - (\mu + \frac{\alpha_0 \sigma}{2})}{\sigma}$$

Διακρίνουμε τις εξής περιπτώσεις:

- Αν  $\sum T_i > B$ , η  $H_0$  απορρίπτεται.
- Αν  $\sum T_i < A$ , η  $H_1$  απορρίπτεται.

Το μέγεθος του δείγματος δίνεται από:

$$N = \inf \left\{ t \in \mathbb{N} : \sum_{i=1}^t T_i \notin [A, B] \right\}$$

Βασιζόμενοι στο SPRT, λαμβάνουμε το κλασικό διάγραμμα CUSUM, θέτοντας  $A = 0$ ,  $h = \frac{B}{\alpha_0}$  και  $k = \frac{\alpha_0}{2}$ . Έτσι, έχουμε:

$$N = \inf \left\{ t \in \mathbb{N} : \sum_{i=1}^t (X_i - \mu - k\sigma) \notin [0, h\sigma] \right\}$$

Ο έλεγχος SPRT, συνεχίζεται έως ότου απορριφθεί η μηδενική υπόθεση. Η πρώτη απόρριψη της  $H_0$  δίνει και την πρώτη ένδειξη στο διάγραμμα CUSUM. Αυτό γίνεται στο σημείο:

$$N_1 = \inf \left\{ t \in \mathbb{N} : \sum_{i=1}^t (X_i - \mu - k\sigma) \notin [0, h\sigma] \right\}$$

Σύμφωνα με τα παραπάνω, αν  $\sum_{i=1}^t (X_i - \mu - k\sigma) > h\sigma$ , τότε η  $H_0$  απορρίπτεται και το CUSUM μας δίνει ένδειξη ότι ο μέσος μετατοπίστηκε από  $\mu$  στην τιμή  $\mu + \alpha\sigma$ . Αντιθέτως, αν  $\sum_{i=1}^t (X_i - \mu - k\sigma) < 0$ , τότε δεν υπάρχει ένδειξη για μετατόπιση, και ξεκινάει νέος έλεγχος SPRT από το χρονικό σημείο  $N_1 + 1$ .

Η δεύτερη απόρριψη της  $H_0$  συμβαίνει στο:

$$N_2 = \inf \left\{ t > N_1 : \sum_{i=N_1+1}^t (X_i - \mu - k\sigma) \notin [0, h\sigma] \right\}$$

Συνεχίζεται αυτή η διαδικασία και κάθε φορά σταματάει με την πρώτη απόρριψη της  $H_0$ . Με άλλα λόγια, προσθέτουμε διαδοχικά τα ποσά  $X_i - \mu - k\sigma$ . Εδώ πρέπει να λάβουμε υπόψη ότι το μέρος του αθροίσματος το οποίο είναι αρνητικό, διαγράφεται.

Έτσι, θεωρούμε τα εξής συσσωρευμένα αθροίσματα:

$$\begin{aligned} S_0^+ &= 0 \\ S_t^+ &= \max\{0, S_{t-1}^+ + (X_t - \mu - k\sigma)\} \text{ για } t \geq 1. \end{aligned}$$

Το μήκος ροής για το CUSUM δίνεται από:

$$N_c^+ = \inf\{t \in \mathbb{N} : S_t^+ > h\sigma\}$$

Όμοια, αν  $\alpha_0 < 0$  προκύπτει:

$$\begin{aligned} S_0^- &= 0 \\ S_t^- &= \min\{0, S_{t-1}^- + (X_t - \mu + k\sigma)\} \text{ για } t \geq 1. \end{aligned}$$

Έστω ότι  $N_c^-$  δηλώνει τον χρόνο διακοπής.

$$N_c^- = \inf\{t \in \mathbb{N} : S_t^- < -h\sigma\}$$

Αν το πρόσημο της μετατόπισης είναι άγνωστο, τότε συνδυάζονται τα δύο μονόπλευρα CUSUM. Το νέο CUSUM αυτό, θα δώσει ένδειξη για εκτός ελέγχου διεργασία τη στιγμή  $t$  αν  $S_t^+ > h\sigma$  ή  $S_t^- < -h\sigma$ . Το νέο μήκος ροής θα είναι  $N_c = \min\{N_c^+, N_c^-\}$ .

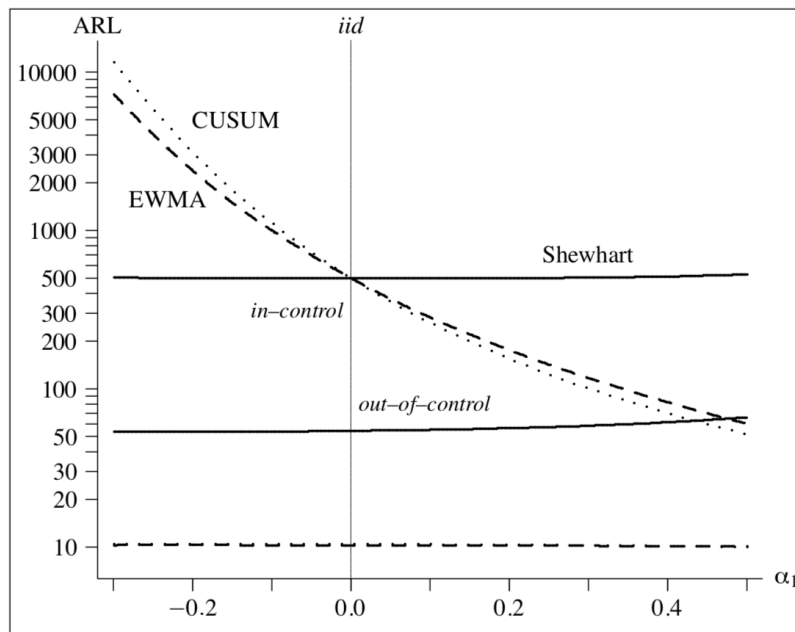
Η παράμετρος  $k$  ονομάζεται τιμή αναφοράς, ενώ το  $h$  καλείται διάστημα απόφασης. Γενικά, η τιμή του  $\alpha_0$  και άρα του  $k$  δεν είναι γνωστή. Αν ενδιαφερόμαστε να ανιχνεύσουμε μεγάλες μετατοπίσεις, τότε επιλέγουμε μεγάλες τιμές  $k$  και  $\alpha_0$ . Αντιθέτως, αν θέλουμε να ανιχνεύσουμε μικρές μετατοπίσεις, το  $k$  επιλέγεται να είναι επίσης μικρό. Προσομοιώσεις έδειξαν ότι για την ανίχνευση μιας μετατόπισης  $\alpha_0\sigma$ , η κατάλληλη επιλογή για το  $k$  είναι η  $k = \frac{|a_0|}{2}$ . Το  $h$  επιλέγεται έτσι ώστε το  $ARL_0$  να είναι ίσο με μία ποσότητα  $\xi$  (π.χ.  $\xi = 500$  ή  $\xi = 370$ ). Φαίνεται πως ο προσδιορισμός των παραμέτρων για το CUSUM καθίσταται πιο εύκολος απ' ό,τι στο EWMA.

**Παρατήρηση:** Τα EWMA και CUSUM προτιμώνται όταν έχουμε μικρές μετατοπίσεις. Για μεγάλες μετατοπίσεις, συνήθως χρησιμοποιούμε το διάγραμμα Shewhart.

### 3.3 Απόδοση των διαγραμμάτων σε εξαρτημένες παρατηρήσεις

Τις περισσότερες φορές, τα δεδομένα που μελετάμε είναι εξαρτημένα μεταξύ τους. Στην περίπτωση αυτή λοιπόν, τα κλασικά διαγράμματα Shewhart, CUSUM και EWMA που περιγράφηκαν, αποδεικνύεται ότι δεν έχουν τόσο καλή απόδοση, και πολλές φορές έχουμε εσφαλμένες ενδείξεις.

Έστω  $\alpha_1 = \text{Corr}(Y_t, Y_{t-1})$ . Παρακάτω παρουσιάζεται ένα διάγραμμα του ARL συναρτήσεως του  $\alpha_1$  για τα διαγράμματα CUSUM, EWMA και Shewhart.



Σχήμα 3.1: Τιμές ARL του διαγράμματος Shewhart (συνεχόμενη γραμμή), EWMA (διακεκομμένη γραμμή) και CUSUM (διακεκομμένη γραμμή με τελείες) παρουσία αυτοσυσχέτισης (μοντέλο (1) με  $m = 1$ ,  $\alpha = 0$  για τα πάνω όρια, και  $\alpha = 1$  για τα κάτω όρια).

Παρατηρούμε ότι το διάγραμμα Shewhart είναι πιο ανθεκτικό σε σχέση με τα EWMA και CUSUM τα οποία φαίνεται να παρουσιάζουν εξαιρετική ευαισθησία στην παρουσία συσχετίσεων.

Για αρνητικές τιμές του  $\alpha_1$  φαίνεται ότι το  $ARL_0$  και στα δύο διαγράμματα είναι μεγαλύτερο απ' ό τι θα έπρεπε, ενώ είναι μικρότερο για θετικές τιμές. Ήδη για  $\alpha_1 = -0.1$  η σχετική απόκλιση από την τιμή  $ARL_0 = 500$  (in-control για ανεξάρτητες-iid μεταβλητές) είναι  $\simeq 100.6\%$  για το EWMA και  $\simeq 123.9\%$  για το CUSUM. Για θετικές τιμές  $\alpha_1$ , παρ'όλο που οι αποκλίσεις δεν είναι τόσο μεγάλες όσο είναι για  $\alpha_1 < 0$ , φτάνουν το 43.4% και 47.6%. Εδώ μπορούμε να πούμε ότι το CUSUM φαίνεται να είναι πιο ευαίσθητο από το EWMA.

Οι αποκλίσεις από το  $ARL_1 \simeq 10$  (out-of-control για iid μεταβλητές) για τα διαγράμματα CUSUM και EWMA φαίνεται να είναι πολύ μικρές, και μάλιστα αν παρατηρήσουμε καλά οι δύο γραμμές σχεδόν ταυτίζονται.

### 3.4 Διαγράμματα του SPC για εξαρτημένες παρατηρήσεις

Όπως είδαμε στην προηγούμενη παράγραφο, τα κλασικά διαγράμματα ελέγχου δεν έχουν καλή απόδοση στην περίπτωση που έχουμε αυτοσυσχετίσεις. Για τον λόγο αυτό, θα παρουσιάσουμε παρακάτω κάποιες εναλλακτικές των διαγραμμάτων αυτών.

#### 3.4.1 Διαγράμματα υπολοίπων (Residual Charts)

Τα διαγράμματα υπολοίπων μπορούν να θεωρηθούν ένα είδος διαγραμμάτων ελέγχου για χρονοσειρές. Βρίσκοντας το μοντέλο που περιγράφει κατάλληλα τη χρονοσειρά μας, είμαστε σε θέση να λάβουμε τις τιμές των υπολοίπων-residuals. Αν αυτά θεωρούνται στατιστικά ανεξάρτητα μεταξύ τους και ακολουθούν την κανονική κατανομή, μπορούν να εφαρμοστούν τα κλασικά διαγράμματα ελέγχου του SPC. Δηλαδή, μετασχηματίζουμε κατάλληλα την αρχική μας χρονοσειρά, εφαρμόζουμε το κατάλληλο μοντέλο και έπειτα ελέγχουμε αν τα υπόλοιπα μπορούν να θεωρηθούν white noise. Αν ναι, τότε είμαστε σε θέση να εφαρμόσουμε τα διαγράμματα CUSUM και EWMA για τα υπόλοιπα, όπως περιγράφηκαν στην αρχή του κεφαλαίου.

Έστω ότι ισχύει η (1). Τότε τα υπόλοιπα θα είναι:

$$\hat{\varepsilon}_t = X_t - \hat{X}_t, \quad t \geq 1$$

όπου  $X_t$  η πραγματική παρατηρούμενη τιμή και  $\hat{X}_t$  η προβλεπόμενη τιμή με βάση το επιλεγμένο μοντέλο. Τα υπόλοιπα υπολογίζονται κάτω από την υπόθεση ότι η διεργασία είναι εντός ελέγχου.

Αν έχουμε ένα  $AR(1)$  μοντέλο με μέσο  $\mu$ , τότε  $\hat{X}_t = \mu + \phi_1 X_{t-1}$ , για  $t \geq 2$  και  $\hat{X}_1 = \mu$ . Άρα προκύπτει:

$$\hat{\varepsilon}_t = X_t - \mu - \phi_1 X_{t-1}, \quad t \geq 2 \quad \text{και} \quad \hat{\varepsilon}_1 = X_1 - \mu$$

Επειδή  $Var(\hat{\varepsilon}_1) = \frac{\sigma^2}{(1 - \phi_1^2)} \geq \sigma^2 = Var(\hat{\varepsilon}_t)$  για  $t \geq 2$ , τα υπόλοιπα πρέπει να κανονικοποιηθούν. Θα έχουμε:

$$\hat{\Delta}_t = \frac{X_t - \mu - \phi_1 X_{t-1}}{\sigma}, \quad t \geq 2$$

$$\hat{\Delta}_1 = \frac{\sqrt{1 - \phi_1^2}(X_1 - \mu)}{\sigma}$$

Όταν η διεργασία είναι υπό έλεγχο, οι τ.μ.  $\hat{\Delta}_t$  είναι ανεξάρτητες και ισόνομες. Αυτό σημαίνει ότι μπορούν να εφαρμοστούν τα διαγράμματα του κεφαλαίου 3 κατευθείαν στα κανονικοποιημένα υπόλοιπα  $\{\hat{\Delta}_t\}$ .

Αν η διεργασία είναι από την αρχή εκτός ελέγχου ( $m = 1$ ), τότε από την (1) έχουμε  $E(X_t) = \mu + \alpha\sqrt{\gamma_0}$  για  $t \geq 1$  και

$$E(\hat{\Delta}_t) = \alpha\sqrt{\frac{1 - \phi_1}{1 + \phi_1}}, \quad \text{για } t \geq 2$$

$$E(\hat{\Delta}_1) = \alpha$$

Αν  $\phi_1 \geq 0$  τότε  $E(\hat{\Delta}_t) \leq \alpha$ , ενώ για  $\phi_1 \leq 0$  έχουμε  $E(\hat{\Delta}_t) \geq \alpha$ . Αυτό σημαίνει ότι για θετικούς συντελεστές  $\phi_1$ , η επιρροή της μετατόπισης  $\alpha\sqrt{\gamma_0}$  μειώνεται, ενώ για αρνητικούς συντελεστές  $\phi_1$ , η επιρροή αυξάνεται.



### 3.4.2 Τροποποιημένο διάγραμμα EWMA

Ο Schmid (1997) εισήγαγε το τροποποιημένο διάγραμμα EWMA, το οποίο βασίζεται στην εξής αναδρομική σχέση:

$$Z_t = (1 - \lambda)Z_{t-1} + \lambda X_t, \text{ για } t \geq 1$$

Έστω  $Z_0 = z_0$  και  $\lambda \in (0, 1]$ . Επίσης υποθέτουμε ότι η διαδικασία-στόχος  $\{Y_t\}$  είναι στάσιμη με μέσο  $\mu$  και συνάρτηση αυτοσυσχέτισης (autocovariance)  $\{\gamma_v\}$ .

**Λήμμα:** Έστω  $\{Y_t\}$  στάσιμη διεργασία με μέσο  $\mu$  και συνάρτηση αυτοσυσχέτισης (autocovariance)  $\{\gamma_v\}$ . Επίσης θεωρούμε την πιο γενική μορφή της σχέσης (1) η οποία είναι η εξής:

$$X_t = Y_t + \sqrt{\gamma_0} f_t(\alpha, m) I_{\{m, m+1, \dots\}}(t)$$

Η συνάρτηση  $f_t(\alpha, m)$  περιγράφει την επιρροή που έχει στην διεργασία η μετατόπιση που προκαλείται και εξαρτάται από την χρονική στιγμή  $t$ , την παράμετρο  $\alpha$  και την τιμή του  $m$ .

α) Αν  $f_t(\alpha, m) = \alpha g_t(m)$ , τότε για κάθε  $t \geq m$  ισχύει:

$$E(Z_t) = \mu + (1 - \lambda)^t (z_0 - \mu) + \lambda \sum_{j=0}^{t-m} (1 - \lambda)^j g_{t-j}(m)$$

β) Αν η  $\{\gamma_v\}$  συγκλίνει απόλυτα ( $\{\gamma_v\} \in l_1$ ):

$$\begin{aligned} \sigma_{e,t}^2 := \text{Var}(Z_t) &= \lambda^2 \sum_{|i| \leq t-1} \gamma_i \sum_{j=\max\{0, -i\}}^{\min\{t-1, t-1-i\}} (1 - \lambda)^{2j+i} \rightarrow \\ &= \frac{\lambda}{2 - \lambda} \sum_{j=-\infty}^{\infty} \gamma_j (1 - \lambda)^{|j|} = \sigma_e^2 \end{aligned}$$

για  $t \rightarrow \infty$ .

γ)

$$Cov(Z_t, Z_{t+h}) = \lambda^2 \sum_{i=-(t+h-1)}^{t-1} \gamma_i \sum_{\tau=\max\{0, -h-i\}}^{\min\{t-1-i, t-1\}} (1-\lambda)^{2\tau+i+h}$$

Έστω ότι έχουμε ένα μοντέλο AR(1). Τότε συνεπάγεται ότι:

$$\sigma_e^2 = \sigma^2 \frac{\lambda}{2-\lambda} \frac{1+\phi_1(1-\lambda)}{1-\phi_1(1-\lambda)}$$

Θέτωντας  $\phi_1 = 0$ , τότε λαμβάνουμε τη διασπορά για ανεξάρτητες παρατηρήσεις. Αν  $f$  είναι η φασματική πυκνότητα πιθανότητας της  $\{Y_t\}$ , τότε έχουμε:

$$\sigma_e^2 = \lambda^2 \int_{-\pi}^{\pi} \frac{f(x)}{1-2(1-\lambda)\cos x + (1-\lambda)^2} dx$$

Συμπεραίνουμε λοιπόν ότι, η διεργασία θα είναι εκτός ελέγχου τη στιγμή  $t$  αν:

$$|Z_t - \mu| > L\sigma_e \Leftrightarrow Z_t > \mu + L\sigma_e \text{ ή } Z_t < \mu - L\sigma_e$$

όπου  $L$  το εύρος των ορίων. Χρησιμοποιήσαμε την  $\sigma_e$  και όχι την  $\sigma_{e,t}$  για να έχουμε ευκολότερους υπολογισμούς και εξομαλυμένα όρια. Αν  $\{\gamma_v\} = 0, \forall v \neq 0$ , επανερχόμαστε στο κλασικό διάγραμμα EWMA. Αν  $\lambda = 1$ , τότε λαμβάνουμε το τροποποιημένο διάγραμμα Shewhart.

Οι Schmid και Schöne (1997) σύγκριναν το μήκος ροής του EWMA για εξαρτημένες και ανεξάρτητες μεταβλητές. Μελέτησαν ένα μονόπλευρο τροποποιημένο διάγραμμα ελέγχου EWMA, του οποίου το μήκος ροής δίνεται από:

$$N_e := \inf\{t \in \mathbb{N} : Z_t - \mu > L\sqrt{Var(Z_t)}\}$$

Σημείωση: Εδώ χρησιμοποιήθηκε η διασπορά  $\sigma_{e,t}^2 = Var(Z_t)$  και όχι η  $\sigma_e$ .

## Θεώρημα

Έστω  $\{Y_t\}$  μία στάσιμη γκαουσιανή διεργασία με μέση τιμή  $\mu$  και συνάρτηση αυτοcovariance  $\{\gamma_v\}$  με  $\gamma_v \geq 0$ ,  $\forall v$ . Έστω  $z_0 = \mu$ . Τότε για κάθε  $k \in \mathbb{Z}$  ισχύει ότι:

$$P(N_e > k) \geq P_{iid}(N_e > k)$$

Απόδειξη: Βλέπε [29].

Παρατήρηση: Μια γκαουσιανή διεργασία είναι μια στοχαστική διαδικασία (μια συλλογή τυχαίων μεταβλητών), έτσι ώστε κάθε πεπερασμένη συλλογή αυτών των τυχαίων μεταβλητών να ακολουθεί πολυμεταβλητή κανονική κατανομή, δηλαδή κάθε πεπερασμένος γραμμικός συνδυασμός αυτών κατανέμεται κανονικά.

Ο δείκτης  $\{iid\}$  σημαίνει ότι οι παρατηρήσεις της διεργασίας  $\{Y_t\}$  θεωρούνται ανεξάρτητες και ισόνομες με μέση τιμή  $\mu$  και διασπορά  $\gamma_0$ . Αυτό βέβαια δε συνεπάγεται απαραίτητα ότι και τα  $\{Z_t\}$  θα είναι ανεξάρτητα μεταξύ τους. Σημειώνουμε εδώ επίσης, ότι το παραπάνω Θεώρημα δεν ισχύει αν  $\gamma_v < 0$ .

Έτσι σύμφωνα με αυτά που προηγήθηκαν, το ARL του EWMA στην περίπτωση διεργασιών αυτοσυσχετισμένων παρατηρήσεων είναι μεγαλύτερο από το ARL του EWMA για ανεξάρτητες παρατηρήσεις, αρκεί  $\gamma_v \geq 0$ . Άρα αν κάποιος θεωρήσει εσφαλμένα ότι οι παρατηρήσεις είναι ανεξάρτητες, τότε το ARL θα υπολογίζεται μικρότερο απ' ότι είναι στην πραγματικότητα.

Γενικά η κατασκευή των τροποποιημένων διαγραμμάτων θεωρείται πιο απλή από τα διαγράμματα υπολοίπων. Το βασικό πρόβλημα έγκειται στον προσδιορισμό της μεταβλητής  $L$ , η οποία ορίζεται για συγκεκριμένη τιμή  $ARL_0$ . Επειδή δεν υπάρχει κάποιος κλειστός τύπος υπολογισμού, κρίνεται απαραίτητη η χρήση προσομοιώσεων. Προφανώς, ακόμα και την περίπτωση των ανεξάρτητων παρατηρήσεων δεν υπάρχει βέλτιστη επιλογή της παραμέτρου  $\lambda$ , η οποία να δίνει το μικρότερο  $ARL_1$  για όλες τις μετατοπίσεις. Αν το μέγεθος της μετατόπισης είναι γνωστό, τότε μπορούμε να επιλέξουμε το  $\lambda$  έτσι ώστε να ελαχιστοποιείται το  $ARL_1$ . Στον Schmid(1997a) μπορούμε να δούμε τους πίνακες με τις τιμές των παραμέτρων για διαφορετικά μεγέθη μετατοπίσεων.

### 3.4.3 Τροποποιημένο διάγραμμα CUSUM

Στην βιβλιογραφία, έχουν γίνει διάφορες προσπάθειες επέκτασης του κλασικού διαγράμματος CUSUM σε αυτοσυσχετισμένα δεδομένα. Για παράδειγμα, ο Nikiforov (1983) εφάρμοσε τον έλεγχο SPRT σε διαδικασίες ARMA με θόρυβο IID. Ο Yashchin (1993) σύγκρινε ένα κατάλληλα τροποποιημένο CUSUM εφαρμοσμένο πάνω στα αρχικά δεδομένα της χρονοσειράς, με ένα CUSUM εφαρμοσμένο πάνω σε μετασχηματισμένα δεδομένα (παρόμοιο με τα διαγράμματα υπολοίπων).

Εμείς θα παρουσιάσουμε ένα τροποποιημένο διάγραμμα CUSUM που πρότεινε ο Schmid (1997b). Υποθέτουμε ότι για  $t \in \mathbb{N}$  η  $\mathbf{Y}_t = (Y_1, \dots, Y_t)'$  είναι κανονικά κατανοημένη με μέση τιμή  $\boldsymbol{\mu}_t = \mu(1, \dots, 1)'$  και (θετικά ορισμένο) πίνακα συσχέτισης  $C_t$ . Επίσης, υποθέτουμε ότι  $\gamma_0 = \text{Var}(Y_t)$ ,  $\forall t$  (υπόθεση ομοσχεδαστικότητας). Οι τιμές των  $\gamma_0, \mu_t$  και  $C_t$  θεωρούνται γνωστές. Συμβολίζουμε με  $f_0$  και  $f_{\alpha, m}$  την συνάρτηση πυκνότητας πιθανότητας της τ.μ.  $\mathbf{X}_t$  στην περίπτωση που δεν παρατηρείται μετατόπιση ( $H_0$ ) και στην περίπτωση που παρατηρείται μετατόπιση τη στιγμή  $m$  ( $H_1$ ), αντίστοιχα. Έτσι το στατιστικό ελέγχου για τον έλεγχο του λόγου πιθανοφανειών θα δίνεται από:

$$-2 \log \frac{f_0(\mathbf{X}_t)}{\max_{1 \leq m \leq t+1} f_{\alpha, m}(\mathbf{X}_t)} = 2 \max \left\{ 0, \max_{1 \leq m \leq t} \alpha \frac{T_{mt} - \alpha \sqrt{\gamma_0} K_{mt}/2}{\sqrt{\gamma_0}} \right\}$$

όπου  $T_{mt} = \mathbf{e}'_{mt} C_t^{-1} (\mathbf{X}_t - \boldsymbol{\mu}_t)$  και  $K_{mt} = \mathbf{e}'_{mt} C_t^{-1} \mathbf{e}_{mt}$ . Τα  $\mathbf{e}_{mt}$  είναι διανύσματα με μηδενικά στις πρώτες  $m - 1$  γραμμές και άσσους στις υπόλοιπες  $t - m + 1$ .

Η διαδικασία σταματάει τη χρονική στιγμή  $t$  αν:

$$\max_{1 \leq m \leq t} \{T_{mt} - k \sqrt{\gamma_0} K_{mt}, -(T_{mt} + k \sqrt{\gamma_0} K_{mt})\} > h \sqrt{\gamma_0}$$

Στην περίπτωση των ανεξάρτητων μεταβλητών:  $k = \frac{|\alpha|}{2}$ .

Στην πραγματικότητα, η διαδικασία υπολογισμού του χρόνου διακοπής είναι χρονοβόρα. Γι' αυτό το λόγο, μπορούμε να υπολογίσουμε το στατιστικό του «νέου» διαγράμματος CUSUM αναδρομικά ως εξής:

Έστω ότι έχουμε μία  $AR(1)$  διεργασία με μέση τιμή  $\mu$ . Το μήκος ροής θα δίνεται από:

$$N_c = \inf\{t \in \mathbb{N} : S_t^+ > h\sqrt{\gamma_0} \text{ ή } S_t^- < -h\sqrt{\gamma_0}\}$$

με

$$S_t^+ = \max\{0, \epsilon_t - k\sqrt{\gamma_0}, S_{t-1}^+ + (1 - \phi_1)(\epsilon_t - (1 - \phi_1)k\sqrt{\gamma_0})\}, \quad t \geq 2$$

$$S_1^+ = \max\{0, (1 - \phi_1^2)(X_1 - \mu - k\sqrt{\gamma_0})\}$$

$$S_t^- = \min\{0, \epsilon_t + k\sqrt{\gamma_0}, S_{t-1}^- + (1 - \phi_1)(\epsilon_t + (1 - \phi_1)k\sqrt{\gamma_0})\}, \quad t \geq 2$$

$$S_1^- = \min\{0, (1 - \phi_1^2)(X_1 - \mu + k\sqrt{\gamma_0})\}$$

Ο έλεγχος SPRT βασίζεται στη σύγκριση των  $f_0(\mathbf{X}_t)$  με το  $f_{\alpha,1}(\mathbf{X}_t)$ . Θα έχουμε:

$$T_t = \log \frac{f_{\alpha,1}(\mathbf{X}_t)}{f_0(\mathbf{X}_t)} = \alpha \frac{T_{1t} - \alpha\sqrt{\gamma_0}K_{1t}/2}{\sqrt{\gamma_0}}$$

Το τροποποιημένο διάγραμμα CUSUM λοιπόν, θα μας δώσει μία ένδειξη τη χρονική στιγμή  $t$  αν:

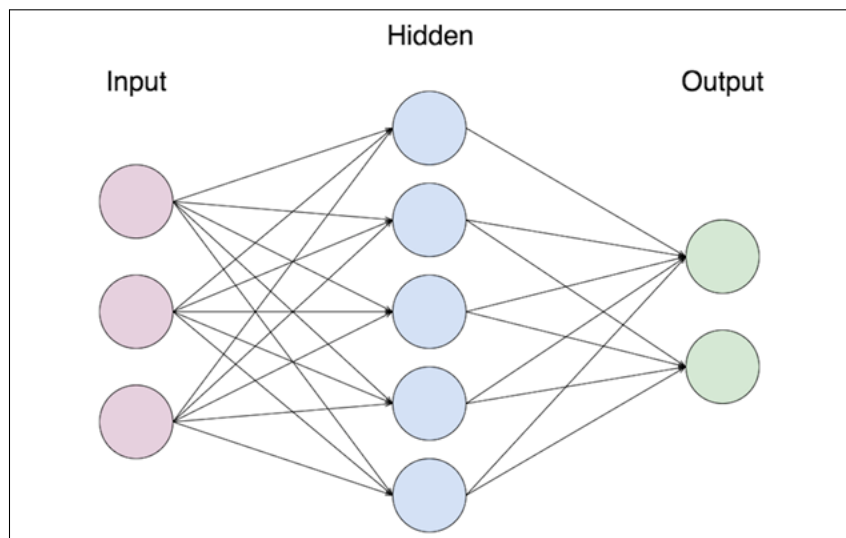
$$\max_{1 \leq m \leq t} \{T_{1m} - k\sqrt{\gamma_0}K_{1m}, -(T_{1m} + k\sqrt{\gamma_0}K_{1m})\} > h\sqrt{\gamma_0}$$

Η τιμή της παραμέτρου  $k$  επιλέγεται όπως και στην περίπτωση των ανεξάρτητων δεδομένων. Η τιμή της παραμέτρου  $h$  καθορίζεται μέσω προσομοιώσεων.

Πίνακες για διάφορες τιμές των παραμέτρων των διαγραμμάτων EWMA και CUSUM μπορούν να βρεθούν στο [11]. Οι τιμές αυτές υπολογίστηκαν μέσω προσομοιώσεων Monte-Carlo για  $ARL_0 = 500$ .

## 4 Τεχνητά νευρωνικά δίκτυα - Artificial neural networks (ANN)

Τα τελευταία χρόνια, το ενδιαφέρον για τον έλεγχο της ποιότητας μιας διεργασίας έχει στραφεί στα τεχνητά νευρωνικά δίκτυα, λόγω της αυξανόμενης ικανότητας τους να μοντελοποιούν συστήματα σε πραγματικό χρόνο. Τα νευρωνικά δίκτυα (NN) κατασκευάζονται από στρώματα συνδεδεμένων μονάδων που ονομάζονται νευρώνες. Ένα «ρηχό δίκτυο» αναφέρεται σε ένα NN με ένα στρώμα εισόδου, ένα στρώμα εξόδου και το πολύ ένα κρυφό στρώμα χωρίς αναδρομική σύνδεση. Καθώς ο αριθμός των στρωμάτων αυξάνεται, η πολυπλοκότητα του δικτύου αυξάνεται επίσης. Μεγαλύτερος αριθμός στρωμάτων ή αναδρομικές συνδέσεις συνεπάγεται και αύξηση του βάθους του δικτύου και του δίνει τη δυνατότητα να παρέχει διάφορα επίπεδα αναπαράστασης δεδομένων και εξαγωγής χαρακτηριστικών, που αναφέρονται ως «βαθιά μάθηση» (deep learning).



Σχήμα 4.1: Τα είδη στρωμάτων του νευρωνικού δικτύου

### Στρώμα εισόδου

Σε κάθε νευρωνικό δίκτυο, το πρώτο στρώμα ονομάζεται στρώμα εισόδου διότι σε αυτό εισάγουμε τα δεδομένα μας,  $\mathbf{x} \in \mathbb{R}^n$ . Ο αριθμός των νευρώνων του ισούται με τη διάσταση  $n$ .

### Στρώμα εξόδου

Σε κάθε νευρωνικό δίκτυο, το τελευταίο στρώμα ονομάζεται στρώμα εξόδου και είναι εκείνο από το οποίο εξάγονται τα αποτελέσματα μας. Ο αριθμός των νευρώνων του εξαρτάται από την φύση του προβλήματος που επιλύουμε.

## Κρυφά στρώματα

Τα ενδιάμεσα στρώματα ονομάζονται κρυφά στρώματα και είναι εκείνα στα οποία γίνεται η επεξεργασία και η ανάλυση των δεδομένων. Ο αριθμός των νευρώνων καθώς και ο αριθμός των στρωμάτων που θα επιλέξουμε δεν είναι κάτι προκαθορισμένο και γι' αυτό θεωρούνται ως υπερπαράμετροι του νευρωνικού δικτύου. Αν τα δεδομένα του προβλήματος είναι γραμμικά διαχωρίσιμα τότε δεν χρειάζεται κανέναν κρυφό στρώμα. Η συνηθέστερη επιλογή είναι 1 ή 2 κρυφά στρώματα, εκτός αν η πολυπλοκότητα του προβλήματος είναι υψηλή οπότε ίσως χρειαστούν 3 – 5. Εδώ πρέπει να σημειώσουμε ότι αν τα κρυφά στρώματα αυξηθούν πολύ (8 – 9) τότε μπορεί να προκύψει το πρόβλημα της υπερπροσαρμογής στα συγκεκριμένα δεδομένα (overfitting). Δηλαδή το νευρωνικό δίκτυο υπερεκπαιδεύεται πάνω στα δεδομένα εισόδου και η ακρίβεια του σε λίγο διαφορετικά δεδομένα προκύπτει να είναι πολύ χαμηλή. Ο αριθμός των νευρώνων σε κάθε κρυφό στρώμα συνήθως είναι ανάμεσα στον αριθμό νευρώνων του στρώματος εισόδου και εξόδου.

Ένας εμπειρικός κανόνας για την επιλογή νευρώνων στα κρυφά στρώματα είναι ο εξής:

$$N_h = \frac{N_s}{\alpha \cdot (N_i + N_o)}$$

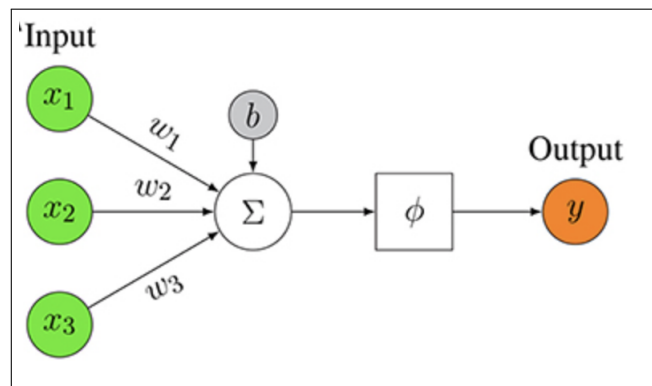
, όπου  $N_i$  ο αριθμός νευρώνων του στρώματος εισόδου,  $N_o$  ο αριθμός νευρώνων στο στρώμα εξόδου,  $N_s$  το πλήθος των δειγμάτων εκπαίδευσης και  $\alpha$  ένας αυθαίρετος παράγοντας συνήθως μεταξύ 2 – 10.

### Υπερπαράμετροι

Ονομάζονται οι παράμετροι του μοντέλου που δεν μπορούν να εκτιμηθούν από τα δεδομένα μας αλλά καθορίζουν την απόδοση του αλγορίθμου. Δηλαδή δεν υπάρχει κάποιος κλειστός τύπος υπολογισμού για αυτές και εξαρτώνται από τη φύση του προβλήματος που έχουμε να επιλύσουμε.

## 4.1 Λειτουργία νευρωνικών δικτύων

Το βασικότερο στοιχείο ενός νευρωνικού δικτύου είναι ο νευρώνας, τον οποίο εισήγαγε ως έννοια ο Rosenblatt (1958). Η κεντρική ιδέα για τη λειτουργία του είναι ότι τα δεδομένα εισόδου,  $\mathbf{x}$ , μαζί με μια μεροληψία  $b$ , σταθμίζονται με βάρη  $\mathbf{w}$  και έπειτα αθροίζονται. Η μεροληψία  $b$ , είναι ένα βαθμωτό μέγεθος, ενώ η είσοδος  $\mathbf{x}$  και τα βάρη  $\mathbf{w}$  είναι διανύσματα, δηλαδή,  $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$  με  $n \in \mathbb{N}$  να αντιστοιχεί στη διάσταση της εισόδου. Το άθροισμα αυτών των όρων, δηλαδή  $z = \mathbf{w}^T \mathbf{x} + b$ , εισέρχεται σε μία συνάρτηση ενεργοποίησης,  $\phi$ , και το αποτέλεσμα αποτελεί δεδομένο εισόδου για το επόμενο στρώμα ή την τελική έξοδο του νευρωνικού. Να σημειώσουμε εδώ ότι η συνάρτηση ενεργοποίησης  $\phi$  μπορεί να διαφέρει για τα κρυφά στρώματα και το στρώμα εξόδου.



Σχήμα 4.2: Εποπτική περιγραφή της λειτουργίας του νευρώνα

**Σημείωση:** Ο όρος μεροληψίας μερικές φορές μπορεί να παραλείπεται.

### Συνάρτηση ενεργοποίησης

Η συνάρτηση ενεργοποίησης  $\phi$  σε ένα νευρωνικό δίκτυο καθορίζει πώς το σταθμισμένο άθροισμα της εισόδου μετατρέπεται σε έξοδο από έναν κόμβο ή κόμβους ενός στρώματος του δικτύου. Συνήθως εκτελεί έναν μη γραμμικό μετασχηματισμό του  $z$ . Η επιλογή της συνάρτησης  $\phi$ , έχει μεγάλο αντίκτυπο στην ικανότητα και την απόδοση του δικτύου. Όλα τα κρυφά στρώματα χρησιμοποιούν συνήθως την ίδια  $\phi$ . Το στρώμα εξόδου, τις περισσότερες φορές, χρησιμοποιεί μια διαφορετική συνάρτηση ενεργοποίησης και εξαρτάται από τον τύπο του προβλήματος που θέλουμε να επιλύσουμε. Παρ' όλο που υπάρχουν πολλές διαφορετικές συναρτήσεις ενεργοποίησης, μόνο ένας μικρός αριθμός από αυτές χρησιμοποιούνται στην πράξη.

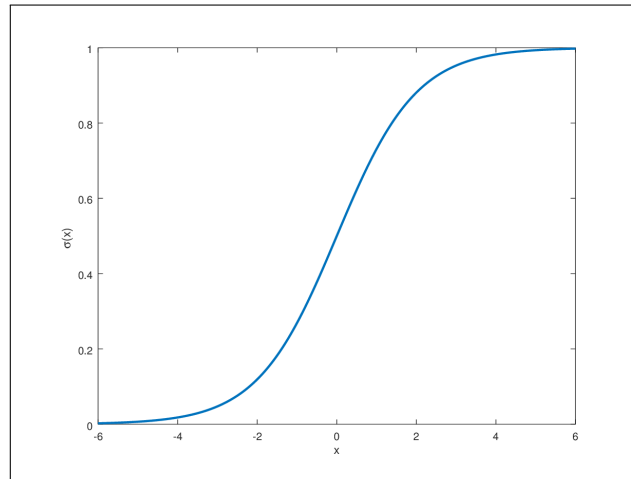
Παρακάτω θα περιγράψουμε συνοπτικά μερικές γνωστές συναρτήσεις ενεργοποίησης.



## Λογιστική σιγμοειδής - Logistic Sigmoid

Η λογιστική σιγμοειδής συνάρτηση ενεργοποίησης ορίζεται ως:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

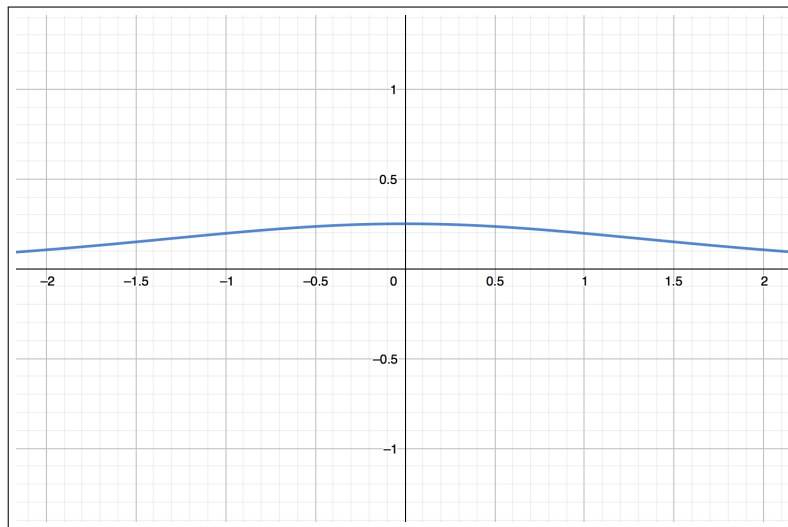


Σχήμα 4.3: Λογιστική σιγμοειδής συνάρτηση ενεργοποίησης

Χαρακτηριστικό της συνάρτησης αυτής είναι ότι αντιστοιχίζει οποιαδήποτε τιμή σε έναν αριθμό που ανήκει στο διάστημα  $[0, 1]$ . Έτσι, ο λόγος για τον οποίο συνήθως χρησιμοποιείται στο στρώμα εξόδου των νευρωνικών δικτύων είναι διότι μπορεί να εκφράσει οποιοδήποτε αριθμό ως πιθανότητα και το αποτέλεσμα μας να είναι πιο εύκολα ερμηνεύσιμο. Η σιγμοειδής συνάρτηση χρησιμοποιείται συνήθως για δυαδική ταξινόμηση στοιχείων (binary classification).

Η λογιστική σιγμοειδής συνάρτηση είναι παραγωγίσιμη και άρα μπορεί να εφαρμοστεί η μέθοδος του backward propagation που θα περιγράψουμε παρακάτω. Συγκεκριμένα η παράγωγος ορίζεται ως:

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$$



Σχήμα 4.4: Παράγωγος της λογιστικής σιγμοειδούς συνάρτησης ενεργοποίησης

Παρατηρούμε ότι παρόλο που η παράγωγος δεν μηδενίζεται τείνει να λαμβάνει αρκετά μικρές τιμές στα άκρα οπότε δημιουργείται το πρόβλημα εξαφανιζόμενων παραγώγων (vanishing gradient).

### Vanishing gradient problem

Κατά τη διάρκεια του backpropagation υπολογίζονται οι παράγωγοι του δικτύου από στρώμα σε στρώμα και πολλαπλασιάζονται σύμφωνα με τον κανόνα της αλυσίδας (από το τελικό επίπεδο στο αρχικό) για τον υπολογισμό των παραγώγων των αρχικών στρωμάτων. Ο λόγος για τον οποίο γίνεται αυτό είναι για να ενημερωθούν οι τιμές των βαρών  $w$  και της μεροληψίας  $b$  έτσι ώστε να ελαχιστοποιείται το σφάλμα. Έτσι, αν έχουμε πολύ μικρές παραγώγους (σχεδόν μηδέν), καθώς αυτές πολλαπλασιάζονται μεταξύ τους θα προκύψει εκθετική μείωση. Αυτό έχει σαν αποτέλεσμα οι τιμές των  $w$ ,  $b$  των αρχικών στρωμάτων να μην ενημερώνονται σωστά και το δίκτυο να μην έχει τελικά καλή απόδοση.

## Softmax

Η συνάρτηση ενεργοποίησης Softmax είναι ουσιαστικά μία πιο γενική μορφή της σιγμοειδούς και ορίζεται ως:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

, όπου  $\mathbf{z} \in \mathbb{R}$  το διάνυσμα με τα δεδομένα εισόδου και  $K$  είναι ο αριθμός των κλάσεων στις οποίες θέλουμε να ταξινομήσουμε τα στοιχεία.

Η συνάρτηση Softmax μετατρέπει οποιοδήποτε διάνυσμα πραγματικών αριθμών σε ένα διάνυσμα με τιμές στο  $[0, 1]$ . Δηλαδή δημιουργεί ένα διάνυσμα πιθανοτήτων και επιλέγει την μεγαλύτερη από αυτές. Οι τιμές του διανύσματος αυτού αθροίζουν στη μονάδα. Για το λόγο αυτό συνήθως χρησιμοποιείται στο τελευταίο στρώμα του δικτύου για την ταξινόμηση στοιχείων σε  $K$  κλάσεις ( $K \geq 2$ ).

Ο όρος  $\sum_{j=1}^K e^{z_j}$  είναι ουσιαστικά ο όρος κανονικοποίησης, δηλαδή αυτός εξασφαλίζει ότι το αποτέλεσμα της συνάρτησης θα κυμαίνεται στο  $0 - 1$ .

## Υπερβολική εφαπτομένη

Μία εναλλακτική της σιγμοειδούς συνάρτησης είναι η υπερβολική εφαπτομένη  $\tanh$ . Δίνεται ως εξής:

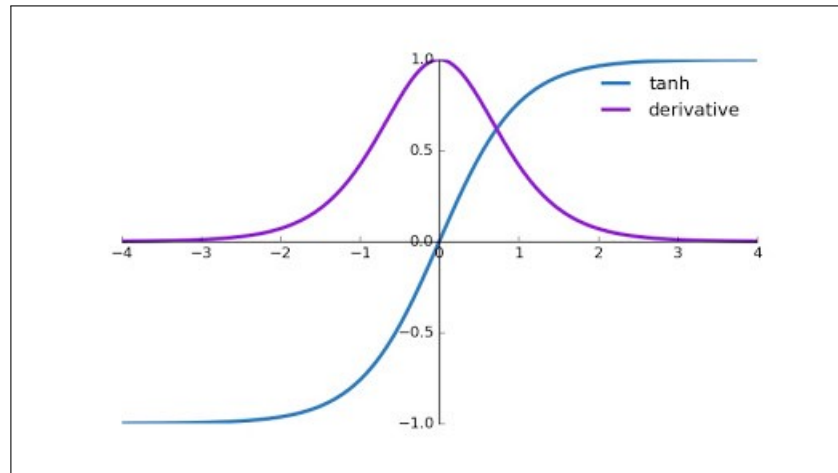
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Οι τιμές που δίνει ανήκουν στο διάστημα  $[-1, 1]$ . Η βασική διαφορά της από την σιγμοειδή συνάρτηση ενεργοποίησης, είναι ότι είναι κεντραρισμένη στο 0.

Η παράγωγος της  $\tanh(x)$  υπολογίζεται ως:

$$\tanh(x)' = \frac{4}{(e^x + e^{-x})^2} = 1 - (\tanh(x))^2$$

και δίνει τιμές στο διάστημα  $(0, 1]$ .



Σχήμα 4.5: Συνάρτηση ενεργοποίησης  $\tanh(x)$

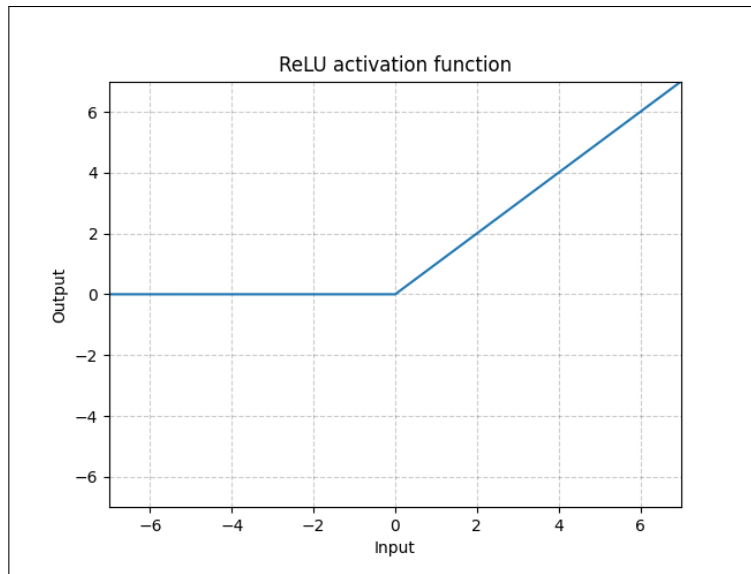
**Σημείωση :** Το κεντράρισμα της συνάρτησης στο 0, βοηθάει ώστε να γίνει πιο γρήγορα η εκπαίδευση του δικτύου καθώς οι παράγωγοι δίνουν πιο απότομη κλίση κατά τη διάρκεια του Backward propagation που θα περιγράψουμε παρακάτω.

Το πρόβλημα εξαφάνισης παραγώγων συνεχίζει να υφίσταται και εδώ.

## ReLU

Η συνάρτηση ReLu είναι από τις πιο δημοφιλείς και πιο αποδοτικές συναρτήσεις ενεργοποίησης και ορίζεται ως εξής:

$$R(x) = \max(0, x)$$



Σχήμα 4.6: Συνάρτηση ενεργοποίησης ReLu

Κάποια από τα σπουδαία πλεονεκτήματα της ReLu είναι τα εξής:

- Γενικά, κατά τη διαδικασία του backpropagation είναι πολύ χρήσιμο να έχουμε συναρτήσεις ενεργοποίησης που συμπεριφέρονται ως γραμμικές ενώ στην πραγματικότητα δεν είναι ώστε να επιτρέπουν ταυτόχρονα περίπλοκες σχέσεις ανάμεσα στα δεδομένα και να καθιστούν τους υπολογισμούς πιο εύκολους. Η συνάρτηση ReLu είναι μία από αυτές, αφού για  $x$  μεγαλύτερο από το 0 ουσιαστικά είναι γραμμική  $R(x) = x, x > 0$  ενώ είναι 0 για τα αρνητικά.
- Επιλύει το πρόβλημα εξαφανιζόμενων παραγώγων αφού η παράγωγος της ορίζεται ως:

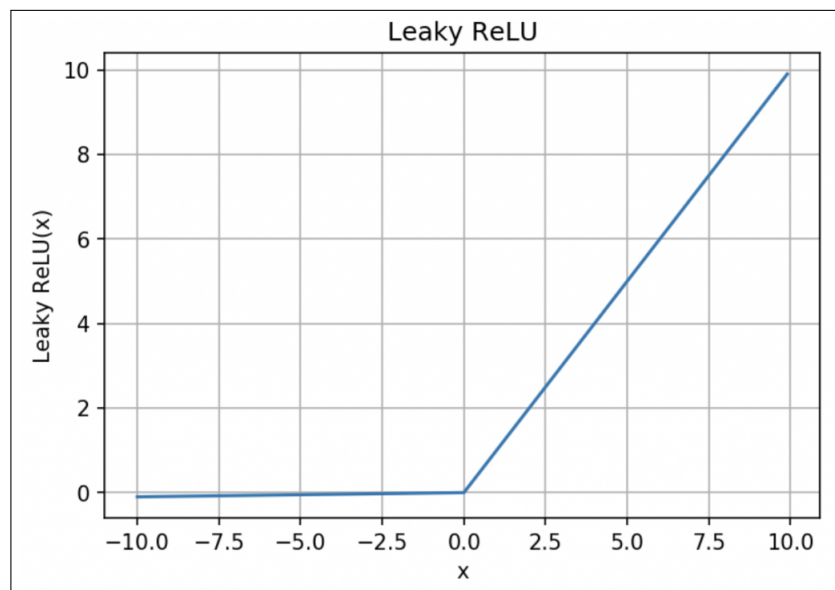
$$R'(x) = \begin{cases} 0 & \text{αν } x < 0 \\ 1 & \text{αν } x > 0 \\ \text{δεν ορίζεται} & \text{αν } x = 0 \end{cases}$$

Βέβαια παρατηρούμε εδώ ότι για αρνητικά  $x$  η παράγωγος λαμβάνει την τιμή 0. Αυτό αποτελεί πρόβλημα στις περιπτώσεις όπου το δίκτυο τελικά δεν εκπαιδεύεται διότι έχει “απενεργοποιηθεί” κάποιος νευρώνας κατά τη διάρκεια του backpropagation, ο οποίος παίζει σημαντικό ρόλο για την ενημέρωση των παραμέτρων. Ένας λόγος για τον οποίο μπορεί να προκύψει το παραπάνω πρόβλημα είναι ο μεγάλος ρυθμός εκμάθησης, τον οποίο θα εξηγήσουμε παρακάτω. Παρ’ όλα αυτά, το φαινόμενο αυτό συμβαίνει σπάνια.

## Leaky ReLu

Η συνάρτηση ενεργοποίησης Leaky ReLu επιλύει το παραπάνω πρόβλημα, θέτωντας αρνητικές τιμές (πολύ κοντά στο 0), όταν  $x < 0$ . Ορίζεται ως εξής:

$$f(x) = \begin{cases} 0.01x & \text{αν } x < 0 \\ x & \text{αν } x > 0 \end{cases}$$

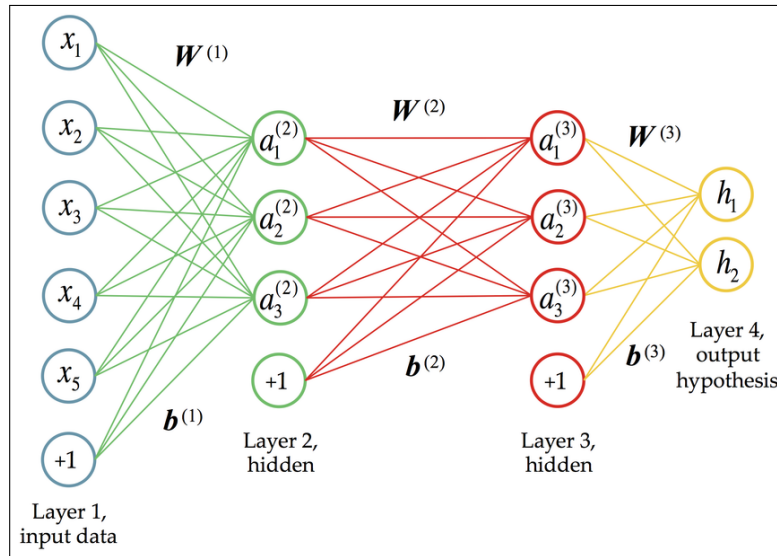


Σχήμα 4.7: Συνάρτηση ενεργοποίησης Leaky ReLu

Επίσης, έχει αποδειχθεί ότι η συνάρτηση Leaky ReLu επιταχύνει ελαφρώς την εκπαίδευση του νευρωνικού. Η παράγωγος της θα είναι:

$$f'(x) = \begin{cases} 0.01 & \text{αν } x < 0 \\ 1 & \text{αν } x > 0 \\ \text{δεν ορίζεται} & \text{αν } x = 0 \end{cases}$$

Παρακάτω φαίνεται μία γραφική αναπαράσταση ενός απλού νευρωνικού δικτύου με 2 κρυφά στρώματα.



Σχήμα 4.8: Νευρωνικό δίκτυο με 2 κρυφά στρώματα

Η συλλογή βαρών μεταξύ στρώματος  $l$  και στρώματος  $l + 1$  δηλώνεται ως  $\mathbf{W}^{(l)} = (w_{jk}^{(l)})$ , όπου είναι το βάρος  $w_{jk}$  που σχετίζεται με τη σύνδεση μεταξύ του κόμβου  $j$  στο στρώμα  $l$  και του κόμβου  $k$  στο στρώμα  $l + 1$ . Οι μεροληψίες για το στρώμα  $l$  δηλώνονται ως  $\mathbf{b}^{(l)} \in \mathbb{R}^{u_l+1}$ . Ο συνολικός αριθμός στρωμάτων (συμπεριλαμβανομένων των στρωμάτων εισόδου και εξόδου) υποδηλώνεται με  $L$ , και ο αριθμός των κρυφών κόμβων στο στρώμα  $l$  υποδηλώνεται ως  $u_l$  (χωρίς τον κόμβο μεροληψίας). Ο κύριος στόχος είναι να βρούμε τα βάρη τα οποία ελαχιστοποιούν μία συνάρτηση κόστους  $J$ .

### 4.1.1 Forward propagation

Στο πρώτο στρώμα εισάγουμε τα δεδομένα μας (κάθε σύνολο δεδομένων έχει 5 χαρακτηριστικά), τα δύο ενδιάμεσα αποτελούν τα κρυφά στρώματα και το τελευταίο προβλέπει τις 2 μεταβλητές απόκρισης. Ο τελευταίος κόμβος σε κάθε επίπεδο εισόδου (+1) αντιπροσωπεύει τον όρο μεροληψίας (bias term). Εδώ ο αριθμός των στρωμάτων είναι  $L = 4$  και ο αριθμός των κόμβων (νευρώνων) σε κάθε επίπεδο είναι  $u_1 = 5, u_2 = 3, u_3 = 3$  και  $u_4 = 2$  (χωρίς τον όρο μεροληψίας). Η τιμή του κρυφού κόμβου  $j$  στο στρώμα  $l \geq 2$  δηλώνεται ως  $\alpha_j^{(l)}$  και ορίζεται ως:

$$\alpha_j^{(l)} = \phi(z_j^{(l)}) \text{ και } z_j^{(l)} = \mathbf{W}_j^{(l-1)} \cdot \boldsymbol{\alpha}^{(l-1)} + b_j^{(l-1)}$$

όπου  $\mathbf{W}_j^{(l-1)}$  είναι η  $j$ -οστή στήλη του πίνακα βαρών  $\mathbf{W}^{(l-1)}$  (δηλαδή όλα τα βάρη που εισέρχονται στον κόμβο  $j$  από το στρώμα  $l - 1$ ),  $\boldsymbol{\alpha}^{(l-1)} \in \mathbb{R}^{u_{l-1}}$  είναι ένα διάνυσμα με όλες τις τιμές των κόμβων στο στρώμα  $l - 1$  (με  $\boldsymbol{\alpha}^{(1)} = x$ , τα αρχικά δεδομένα εισόδου),  $b_j^{(l-1)}$  είναι όρος μεροληψίας του στρώματος  $l - 1$  που προστίθεται στον κόμβο  $j$  και  $\phi$  μία συνάρτηση ενεργοποίησης. Με αυτόν τον τρόπο, τροφοδοτούμε όλα τα στρώματα και τελικά προβλέπουμε τις τιμές  $h_1, h_2$ .

### 4.1.2 Backward propagation

Τώρα σκοπός είναι να δούμε κατά πόσο οι προβλεπόμενες τιμές απέχουν από τις πραγματικές μέσω μιας συνάρτησης κόστους  $J(\mathbf{W}, \mathbf{b})$ . Μερικές γνωστές συναρτήσεις κόστους που χρησιμοποιούνται στα νευρωνικά δίκτυα είναι οι εξής:

- Μέσο Απόλυτο σφάλμα:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

, όπου  $n$  το μέγεθος της εισόδου,  $y_i$  η πραγματική τιμή και  $\hat{y}_i$  η εκτίμηση του δικτύου.

- Μέσο τετραγωνικό σφάλμα:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

, όπου  $n$  το μέγεθος της εισόδου,  $y_i$  η πραγματική τιμή και  $\hat{y}_i$  η εκτίμηση του δικτύου.



- Cross Entropy:

$$J(\mathbf{x}) = - \sum_{i=1}^N p(x_i) \log q(x_i)$$

, όπου  $N$  το μέγεθος των κλάσεων που θέλω να ταξινομήσω τα γεγονότα (δηλαδή το μέγεθος εξόδου),  $p(x_i)$  η πραγματική τιμή και  $q(x_i)$  η εκτίμηση του δικτύου.

Για  $N = 2$ :

- Binary Cross Entropy:

$$J(x) = -[p \log q + (1 - p) \log(1 - q)]$$

, όπου  $p$  η πραγματική τιμή (0 ή 1) και  $q$  η εκτίμηση του δικτύου.

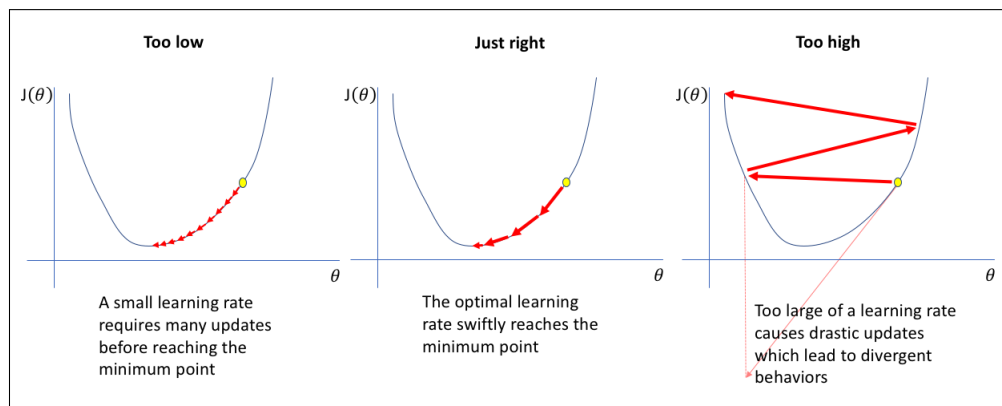
Σκόπος είναι να αναπροσαρμόσουμε τις τιμές των  $\mathbf{W}$  και  $\mathbf{b}$ , υπολογίζοντας τις μερικές παραγώγους της συνάρτησης  $J(\mathbf{W}, \mathbf{b})$  ως προς τα βάρη  $\mathbf{W}$  και τις μεροληψίες  $\mathbf{b}$ , χρησιμοποιώντας κάποιον αλγόριθμο βελτιστοποίησης. Μερικοί γνωστοί αλγόριθμοι είναι οι Batch Gradient Descent, Mini-Batch Gradient Descent, RMSPProp (Root Mean Square Propagation) και Adam (Adaptive moment estimation).

- Batch Gradient Descent

Ο αλγόριθμος Batch Gradient Descent (ή απλά Gradient Descent) είναι ένας από τους πιο δημοφιλείς και απλούς αλγόριθμους για την βελτιστοποίηση των νευρωνικών δικτύων. Είναι ένας επαναληπτικός αλγόριθμος που χρησιμοποιείται για την εύρεση της ελάχιστης τιμής μιας συνάρτησης, λαμβάνοντας υπόψη όλα τα δεδομένα εκπαίδευσης σε κάθε βήμα. Ως δεδομένα εκπαίδευσης ορίζουμε ζεύγη  $(x_i, y_i)$  όπου  $x_i$  δεδομένα εισόδου και  $y_i$  τα αντίστοιχα δεδομένα εξόδου (δοσμένες τιμές-στόχοι) και  $i = 1, 2, \dots, n$ .

## Ρυθμός εκμάθησης (Learning Rate)

Το μέγεθος του βήματος για την επίτευξη του ελαχίστου της συνάρτησης (ταχύτητα σύγκλισης) ονομάζεται ρυθμός εκμάθησης ( $\rho$ ). Το βήμα αυτό δε θα πρέπει να είναι ούτε πολύ μεγάλο αλλά ούτε και πολύ μικρό. Ο λόγος είναι ο εξής: αν ο ρυθμός είναι μεγάλος τότε διατρέχουμε τον κίνδυνο υπέρβασης των ελαχίστων (απόκλιση) ενώ από την άλλη πλευρά, ο μικρός ρυθμός καθιστά τον αλγόριθμο χρονοβόρο. Αυτό παρουσιάζεται στην παρακάτω εικόνα.



Σχήμα 4.9: Ορθή επιλογή του ρυθμού εκμάθησης

Η μέθοδος αποτελείται ουσιαστικά από 2 βήματα:

1. Υπολογισμός της πρώτης παραγώγου της συνάρτησης κόστους  $J$  ως προς όλες τις παραμέτρους  $\theta$  στο τρέχον βήμα.
2. Κίνηση προς την αντίθετη κατεύθυνση της παραγώγου, κατά μία ποσότητα  $\rho \nabla_{\theta} J$ .

Μαθηματικά εκφράζεται ως:

$$\theta(t + 1) = \theta(t) - \rho_t \nabla_{\theta} J(\mathbf{W}, b)$$

, όπου  $\theta$  η παράμετρος ( $\mathbf{W}$  ή  $b$ ),  $\rho_t$  ο ρυθμός εκμάθησης,  $J$  η συνάρτηση κόστους και  $\nabla_{\theta} J(\mathbf{W}, b) = \frac{\partial J}{\partial \theta}$ .

Οι παράγωγοι υπολογίζονται ξεχωριστά για κάθε παράμετρο σε κάθε στρώμα  $l$ , σύμφωνα με τον κανόνα της αλυσίδας:

$$\frac{\partial J}{\partial W^{(l)}} = \frac{\partial J}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial W^{(l)}}$$

και

$$\frac{\partial J}{\partial b^{(l)}} = \frac{\partial J}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial b^{(l)}}$$

Βασικό πλεονέκτημα της μεθόδου είναι ότι για κυρτές συναρτήσεις συγκλίνει σταθερά και σίγουρα στο ολικό ελάχιστο και για μη κυρτές στο τοπικό.

Στις περιπτώσεις όπου η συνάρτηση κόστους είναι κυρτή, μπορούμε να υπολογίσουμε τον βέλτιστο ρυθμό εκμάθησης ως εξής:

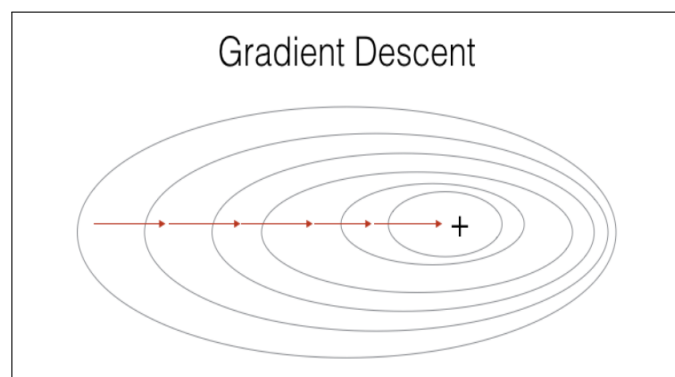
Έστω  $w^*$  η βέλτιστη τιμή του  $w$ :

$$J \approx J(w^*) + \frac{1}{2} \frac{\partial^2 J}{\partial w^2} (w - w^*)^2 \text{ (Taylor)}$$

$$w(t+1) = w(t) - \rho_t \frac{\partial J}{\partial w} = w(t) - \rho_t \frac{\partial^2 J}{\partial w^2} (w - w^*)$$

$$\rho^* = \left( \frac{\partial^2 J}{\partial w^2} \right)^{-1}$$

Όμοια για  $\theta = b$ .



Σχήμα 4.10: Οπτική απεικόνιση της μεθόδου Gradient Descent

Καθώς η παραπάνω μέθοδος αυτή λαμβάνει σε κάθε βήμα όλα τα δεδομένα εκπαίδευσης, προκύπτει να είναι πολύ αργή όταν αυτά είναι πολύ μεγάλα. Μία λύση στο πρόβλημα αυτό είναι η εφαρμογή της μεθόδου Mini Batching Gradient Descent.

- Mini Batching Gradient Descent

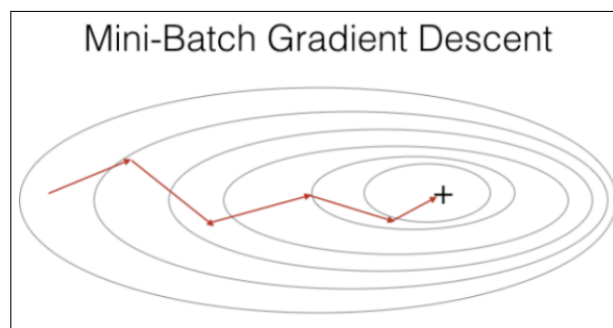
Η βασική διαφορά από την απλή μέθοδο Gradient Descent είναι ότι χωρίζει τα δεδομένα εκπαίδευσης σε μικρότερα σύνολα μεγέθους  $m$ . Δηλαδή για τα δεδομένα  $\mathbf{x} = [x_1, x_2, \dots, x_n]$  και τις τιμές στόχους  $\mathbf{y} = [y_1, y_2, \dots, y_n]$ , παίρνουμε τα υποσύνολα

$$x^{\{1\}} = [x_1, x_2, \dots, x_m], x^{\{2\}} = [x_{m+1}, x_{m+2}, \dots, x_{2m}], \dots, x^{\{\frac{n}{m}\}} = [x_{n-m+1}, \dots, x_n]$$

και αντίστοιχα

$$y^{\{1\}} = [y_1, y_2, \dots, y_m], y^{\{2\}} = [y_{m+1}, y_{m+2}, \dots, y_{2m}], \dots, y^{\{\frac{n}{m}\}} = [y_{n-m+1}, \dots, y_n].$$

Έτσι, σε κάθε βήμα επιλέγουμε σειριακά ένα σετ δεδομένων  $(x^k, y^k)$ ,  $k = 1, 2, \dots, \frac{n}{m}$  και εκτελούμε τον αλγόριθμο Gradient Descent που περιγράψαμε παραπάνω. Με αυτό τον τρόπο επιταχύνουμε την διαδικασία εκπαίδευσης του νευρωνικού δικτύου και μειώνουμε την χωρική πολυπλοκότητα. Παρ' όλα αυτά, εισάγεται μία νέα υπερπαράμετρος  $m$  που πρέπει να οριστεί κατάλληλα και η σύγκλιση δεν γίνεται με σταθερό ρυθμό, αλλά δημιουργεί ταλάντωση ανάμεσα σε υπο-βέλτιστα τοπικά ελάχιστα.



Σχήμα 4.11: Οπτική απεικόνιση της μεθόδου Mini Batch Gradient Descent

- RMSProp (Root Mean Square Propagation)

Η μέθοδος βελτιστοποίησης RMSProp είναι μια αδημοσίευτη μέθοδος που πρότεινε ο Geoff Hinton σε μία διάλεξη του στο Coursera. Οι παράγωγοι των σύνθετων συναρτήσεων που χρησιμοποιούνται στα νευρωνικά δίκτυα έχουν την τάση είτε να εξαφανίζονται (vanishing gradients problem) είτε να εκρήγνυνται (exploding gradients problem).

Η μέθοδος RMSprop επιλύει το παραπάνω πρόβλημα χρησιμοποιώντας έναν κινούμενο μέσο όρο των τετραγώνων των παραγώγων έτσι ώστε να τις κανονικοποιήσει. Αυτή η κανονικοποίηση εξισορροπεί το μέγεθος του βήματος (ρυθμός εκμάθησης), μειώνοντας το για μεγάλες κλίσεις για να αποφευχθεί η έκρηξη και αυξάνοντας το για μικρές κλίσεις για να αποφευχθεί η εξαφάνιση. Η μέθοδος είναι η εξής:

$$g_t = \nabla_w J[w(t-1)]$$

$$\mathbb{E}[g^2]_t = \beta \mathbb{E}[g^2]_{t-1} + (1 - \beta) g_t^2$$

$$w(t+1) = w(t) - \frac{\rho}{\sqrt{\mathbb{E}[g^2]_t + \epsilon}} g_t$$

, όπου  $\beta, \rho$  υπερπαραμέτροι. Ο Geoff Hinton προτείνει ως βέλτιστες επιλογές τις τιμές  $\beta = 0.9$  και  $\rho = 0.001$ .

Όμοια για  $\theta = b$ . Ανάμεσα στα στρώματα τη θέση της συνάρτησης κόστους παίρνει η συνάρτηση ενεργοποίησης  $\phi$ .

- Adam (Adaptive moment estimation)

Ο αλγόριθμος Adam συνδυάζει τα πλεονεκτήματα δύο άλλων γνωστών μεθόδων: RMSprop και AdaGrad. Ο AdaGrad προσαρμόζει το ρυθμό εκμάθησης, επιλέγοντας μικρό ρυθμό για παραμέτρους που σχετίζονται με συχνά εμφανιζόμενα χαρακτηριστικά και μεγάλο ρυθμό για παραμέτρους που σχετίζονται με σπάνια εμφανιζόμενα χαρακτηριστικά. Για αυτόν τον λόγο, είναι κατάλληλος για την επεξεργασία αραιών δεδομένων (περιέχουν πολλές μηδενικές τιμές). Με την έννοια χαρακτηριστικά αναφερόμαστε ουσιαστικά στα δεδομένα εισόδου και στους κόμβους όλων των κρυφών στρωμάτων. Η μέθοδος Adam είναι απλή στην εφαρμογή, υπολογιστικά αποδοτική με μικρή χωρική πολυπλοκότητα και κατάλληλη για προβλήματα με μεγάλο όγκο δεδομένων και / ή παραμέτρων.

Ο αλγόριθμος βελτιστοποίησης Adam ενημερώνει συνεχώς τους εκθετικά κινούμενους μέσους όρους των παραγώγων ( $m_t$ ) και του τετραγώνου των παραγώγων ( $v_t$ ). Άρα ουσιαστικά,  $m_t$  και  $v_t$  είναι οι εκτιμήσεις των πρώτων και δεύτερων ροπών των παραγώγων  $g_t$ . Οι υπερπαραμέτροι  $\beta_1, \beta_2 \in [0, 1)$  ελέγχουν τους ρυθμούς εκθετικής πτώσης αυτών των κινούμενων μέσων. Δίνονται ως εξής:

1. Αρχικοποιούμε  $m_0 = v_0 = t = 0$
2. Όσο δεν έχει επιτευχθεί σύγκλιση, κάνε τα εξής:
3.  $t = t + 1$
4.  $g_t = \nabla_{\theta} f_t[\theta(t-1)]$ , όπου  $f_t = J$  στο τελευταίο στρώμα και  $f_t = \phi$  στα υπόλοιπα στρώματα.
5.  $m_t = \beta_1 m_t + (1 - \beta_1) g_t$
6.  $v_t = \beta_2 v_t + (1 - \beta_2) g_t^2$
7.  $\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$
8.  $\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$
9.  $\theta_t = \theta_{t-1} - \rho \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$

, όπου  $t$  το χρονικό βήμα,  $\epsilon$  μία μικρή σταθερά για αποφυγή διαίρεσης με το 0,  $\rho$  ο ρυθμός εκμάθησης και  $\theta = w$  ή  $\beta$ .

Στα βήματα 7 και 8 γίνεται μία διόρθωση μεροληψίας, καθώς αρχικοποιούμε τα  $m_t, v_t$  στο 0 και οι παράμετροι  $\beta_1, \beta_2$  βρίσκονται κοντά στη μονάδα. Στο βήμα 9 εφαρμόζεται ο τύπος αναπροσαρμογής των παραμέτρων (παρόμοιος με RMSprop).

Μία καλή αρχικοποίηση για τον αλγόριθμο θεωρείται η:  $\rho = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  και  $\epsilon = 10^{-8}$ .

Έχει αποδειχθεί ότι αλγόριθμος Adam τις περισσότερες φορές είναι πιο αποδοτικός από άλλες μεθόδους βελτιστοποίησης και γι' αυτό χρησιμοποιείται πολύ συχνά στα νευρωνικά δίκτυα. Φυσικά, συνεχώς αναπτύσσονται παραλλαγές του Adam όπως ο AdaMax (Kingma and Ba, 2015) στον οποίο χρησιμοποιείται η νόρμα  $l_{\infty}$  για τον υπολογισμό του  $v_t$  και δε χρειάζεται να γίνει διόρθωση μεροληψίας. Ο AMSGrad είναι μία άλλη παραλλαγή του Adam, ο οποίος χρησιμοποιεί το μέγιστο των προηγούμενων  $v_t$  (ή  $\hat{v}_t$ ) αντί για τον εκθετικό μέσο όρο και έτσι προκύπτει ένας μη αυξανόμενος ρυθμός εκμάθησης που επιλύει το πρόβλημα σύγκλισης στη βέλτιστη λύση.

Παρακάτω θα περιγράψουμε την διαδικασία του **Backward Propagation**.

Θεωρούμε την εξής συνάρτηση κόστους:

$$\text{MSE} = J(\mathbf{W}, b) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

, όπου  $\hat{y}_i$  μπορούμε να αντικαταστήσουμε την έκφραση  $\phi_o(\alpha_j^L)$ . Ως  $\phi_o$  ορίζουμε την συνάρτηση ενεργοποίησης στο στρώμα εξόδου (output layer), για παράδειγμα η σιγμοειδής  $\sigma(\cdot)$ . Ο δείκτης  $j$  συμβολίζει τον κόμβο ενώ ο δείκτης  $L$  το τελευταίο στρώμα.

Έπειτα ορίζουμε:

$$\delta_j^{(l)} := \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial z_j^{(l)}}$$

, όπου  $z_j^{(l)}$  είναι η είσοδος στην συνάρτηση ενεργοποίησης του κόμβου  $j$  στο στρώμα  $l$ . Ο όρος αυτός καλείται συνήθως σφάλμα του  $j$ -οστού κόμβου του στρώματος  $l$ .

Για το στρώμα εξόδου προκύπτει ότι:

$$\delta_j^{(L)} = \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial z_j^{(L)}} = \sum_{k=1}^{u_l} \frac{\partial J(\mathbf{W}, b)}{\partial \alpha_k^{(L)}} \frac{\partial \alpha_k^{(L)}}{\partial z_j^{(L)}}$$

Ξέρουμε ότι  $\alpha_j^{(L)} = \phi_o(z_j^{(L)})$ . Οπότε  $\frac{\partial \alpha_k^{(L)}}{\partial z_j^{(L)}} = \phi_o'(z_j^{(L)})$ .

Δηλαδή:

$$\delta_j^{(L)} = \frac{\partial J(\mathbf{W}, b)}{\partial \alpha_j^{(L)}} \phi_o'(z_j^{(L)})$$

Το αποτέλεσμα του όρου  $\frac{\partial J(\mathbf{W}, b)}{\partial \alpha_j^{(L)}}$  εξαρτάται από την συνάρτηση κόστους που χρησιμοποιούμε. Άρα στην περίπτωση της  $MSE$  προκύπτει:

$$\delta_j^{(L)} = [\phi_o(z_j^{(L)}) - y_i] \phi'_o(z_j^{(L)})$$

Για τα ενδιάμεσα στρώματα, μπορούμε να υπολογίσουμε την αναδρομική σχέση των  $\delta_j^{(l)}$ :

$$\delta_j^{(l)} = \frac{\partial J(\mathbf{W}, b)}{\partial z_j^{(L)}} = \sum_{k=1}^{u_l} \frac{\partial J(\mathbf{W}, b)}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}}$$

Όμως,  $\frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} = w_{jk}^{(l+1)} \phi'(z_j^{(l)})$ , όπου  $\phi$  η συνάρτηση ενεργοποίησης των ενδιάμεσων

στρωμάτων. Επίσης  $\frac{\partial J(\mathbf{W}, b)}{\partial z_j^{(l+1)}} = \delta_j^{(l+1)}$ . Άρα προκύπτει:

$$\delta_j^{(l)} = \left[ \sum_{k=1}^{u_{l+1}} w_{jk}^{(l)} \delta_k^{(l+1)} \right] \phi'(z_j^{(l)})$$

Εφαρμόζοντας ξανά τον κανόνα αλυσίδας έχουμε:

$$\frac{\partial J(\mathbf{W}, b)}{\partial w_{jk}^{(l)}} = \sum_{k=1}^{u_l} \frac{\partial J(\mathbf{W}, b)}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial w_{jk}^{(l)}}$$

Ξέρουμε ότι  $\frac{\partial z_k^{(l+1)}}{\partial w_{jk}^{(l)}} = \alpha_j^{(l)}$  και  $\delta_j^{(l)} = \frac{\partial J(\mathbf{W}, b)}{\partial z_j^{(l)}}$ . Άρα τελικά μπορούμε να ορίσουμε:

$$\frac{\partial J(\mathbf{W}, b)}{\partial w_{jk}^{(l)}} = \alpha_j^{(l)} \delta_j^{(l+1)}$$

για  $j = 1, 2, \dots, u_{l+1}$ ,  $k = 1, 2, \dots, u_l + 1$  και  $l = 1, 2, \dots, L$ .



Όμοια για τις μεροληψίες  $\mathbf{b}$  θα έχουμε:

$$\frac{\partial J(\mathbf{W}, b)}{\partial b_k^{(l)}} = \sum_{k=1}^{u_l} \frac{\partial J(\mathbf{W}, b)}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial b_k^{(l)}} = \sum_{k=1}^{u_l} \frac{\partial J(\mathbf{W}, b)}{\partial z_k^{(l+1)}}$$

Άρα τελικά:

$$\boxed{\frac{\partial J(\mathbf{W}, b)}{\partial b_k^{(l)}} = \delta_j^{(l+1)}}$$

Συνοπτικά, οι εξισώσεις για τη μέθοδο του Backpropagation φαίνονται παρακάτω:

1.  $\delta_j^{(L)} = \frac{\partial J(\mathbf{W}, b)}{\partial \alpha_j^{(L)}} \phi'_o(z_j^{(L)})$
2.  $\delta_j^{(l)} = \left[ \sum_{k=1}^{u_{l+1}} w_{jk}^{(l)} \delta_k^{(l+1)} \right] \phi'(z_j^{(l)})$
3.  $\frac{\partial J(\mathbf{W}, b)}{\partial w_{jk}^{(l)}} = \alpha_j^{(l)} \delta_j^{(l+1)}$
4.  $\frac{\partial J(\mathbf{W}, b)}{\partial b_k^{(l)}} = \delta_j^{(l+1)}$

Εφόσον, έχουμε υπολογίσει τις παραγώγους των παραμέτρων είμαστε σε θέση να εφαρμόσουμε έναν αλγόριθμο βελτιστοποίησης από αυτούς που περιγράψαμε έτσι ώστε να ενημερώσουμε τις τιμές των παραμέτρων, ελαχιστοποιώντας τη συνάρτηση κόστους  $J$ . Αφού γίνει αυτό, επανερχόμαστε στη μέθοδο του Forward propagation και επαναλαμβάνουμε τη διαδικασία.

## Εποχή

Τα νευρωνικά δίκτυα εκπαιδεύονται σε εποχές. Ως εποχή ορίζεται μία διαπέραση forward και backward propagation, δηλαδή ένας πλήρης κύκλος εκπαίδευσης. Για να μάθει το νευρωνικό δίκτυο να αποδίδει καλά σε νέα δεδομένα, συνήθως χρειάζεται ένας μεγάλος αριθμός εποχών. Επειδή δεν υπάρχει κάποιος συγκεκριμένος τύπος υπολογισμού, η εποχή αποτελεί μία ακόμη υπερπάρμετρο του δικτύου.

### 4.1.3 Το πρόβλημα της υπερπροσαρμογής

Ένα πρόβλημα που μπορεί να προκύψει κατά την εκπαίδευση των νευρωνικών δικτύων είναι η υπερπροσαρμογή στα δεδομένα εκπαίδευσης (overfitting). Αυτό σημαίνει ότι το μοντέλο εκπαιδεύεται παραπάνω εποχές απ' ότι χρειάζεται στην πραγματικότητα ή ότι η δομή του νευρωνικού είναι υπερβολικά περίπλοκη για τα συγκεκριμένα δεδομένα ή ότι αυτά είναι λίγα και έτσι τα τελικά βάρη προσαρμόζονται υπερβολικά στις μεταβλητές εισόδου με αποτέλεσμα η απόδοση του σε νέα δεδομένα να είναι χαμηλή.

## Μέθοδος Dropout

Το Dropout είναι μια μέθοδος κανονικοποίησης, η οποία μπορεί να αποτρέψει το πρόβλημα της υπερπροσαρμογής. Ο τρόπος που λειτουργεί είναι ο κάτωθι:

Κατά τη διάρκεια της εκπαίδευσης του νευρωνικού δικτύου, ένας αριθμός δεδομένων εξόδου των στρώματων αγνοούνται με τυχαίο τρόπο, δηλαδή γίνονται dropped-out. Τον αριθμό αυτόν, τον ορίζουμε εμείς με το λεγόμενο drop rate, δηλαδή ορίζουμε το ποσοστό των δεδομένων που αφαιρούνται μετά από κάθε στρώμα. Οι μονάδες που γίνονται dropped-out, αφαιρούνται εντελώς από το δίκτυο, μαζί με τις συνδέσεις που είχαν σε άλλα στρώματα. Το Dropout έχει ως αποτέλεσμα να κάνει την εκπαίδευση πιο θορυβώδη, αναγκάζοντας τους κόμβους ενός στρώματος να αναλάβουν πιθανώς περισσότερη ή λιγότερη ευθύνη για τα δεδομένα που λαμβάνουν ως είσοδο. Τέλος, κατά την τεχνική dropout μπορεί να «σπάσουν» συνδέσεις μεταξύ των στρώματων οι οποίες διόρθωναν σφάλματα από τα προηγούμενα στρώματα. Παρ' όλο που η διόρθωση των σφαλμάτων φαίνεται ως κάτι θετικό, δυστυχώς αποτελεί τροχοπέδη για την γενίκευση του δικτύου σε άλλα δεδομένα καθώς ευνοεί την υπερπροσαρμογή στα δεδομένα εκπαίδευσης. Αυτό συμβαίνει καθώς κατά τη διόρθωση των σφαλμάτων μπορεί να δημιουργηθούν περίπλοκες διασυνδέσεις και συν-προσαρμογές οι οποίες κρατάνε στη μνήμη λεπτομέρειες από τα δεδομένα εισόδου και έτσι αποτρέπουν τη γενίκευση του δικτύου. Η μέθοδος dropout λοιπόν, κρίνεται απαραίτητη για την αποφυγή τέτοιου είδους προβλημάτων, καθιστώντας το δίκτυο πιο στιβαρό.

## 4.2 Αναδρομικά νευρωνικά δίκτυα - Recurrent neural networks (RNN)

Μεταξύ των διαφόρων μοντέλων NN, τα αναδρομικά νευρωνικά δίκτυα (RNN), έχουν αποδειχθεί ιδιαίτερα χρήσιμα για τη μοντελοποίηση χρονοσειρών αφού όπως θα δούμε επιτρέπουν τη σύνδεση νευρώνων που βρίσκονται στο ίδιο κρυφό στρώμα. Κάθε ολοκληρωμένο RNN είναι ένας συνδυασμός κελιών RNN. Τα πιο δημοφιλή κελιά RNN που χρησιμοποιούνται για τη μοντελοποίηση χρονοσειρών είναι το κελί Elman RNN (ERNN), το κελί LSTM (Long Short-term Memory) και το κελί GRU (Gated Recurrent Unit).

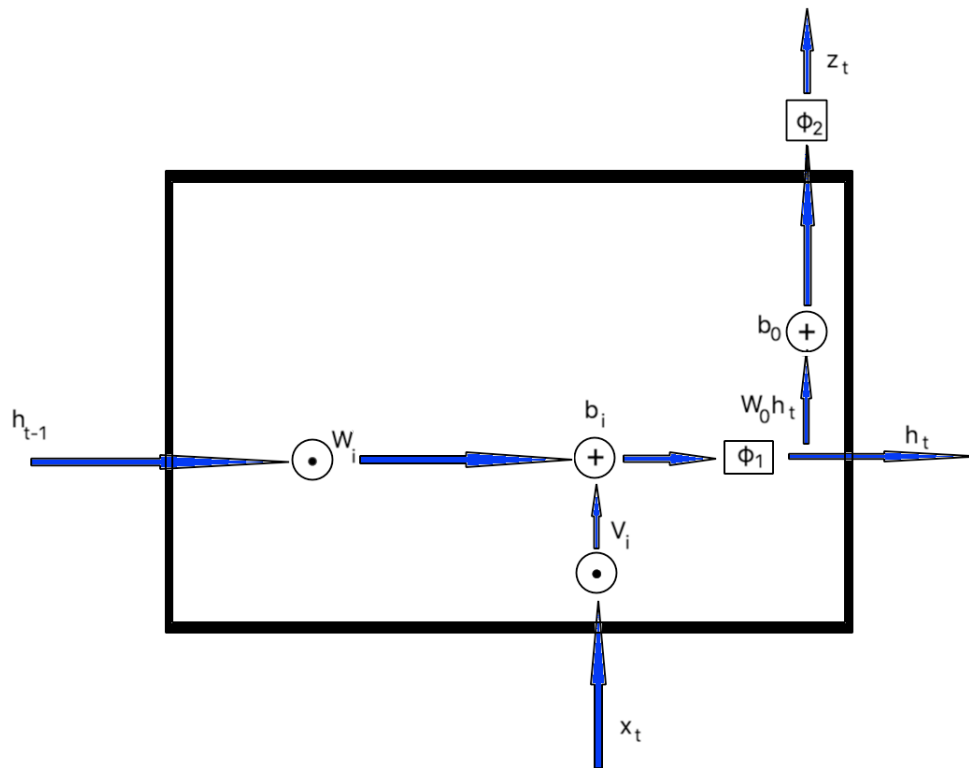
### 4.2.1 ERNN - Elman RNN

Το κελί ERNN αποτελεί το πιο απλό και βασικό κελί RNN το οποίο εισήγαγε ο Elman (1990). Η δομή του παρουσιάζεται παρακάτω:

$$h_t = \phi_1(W_i \cdot h_{t-1} + V_i \cdot x_t + b_i)$$

$$z_t = \phi_2(W_o \cdot h_t + b_o)$$

, όπου  $h_t \in \mathbb{R}^d$  η κρυφή έξοδος του κελιού (δηλαδή πληροφορίες από τα προηγούμενα δεδομένα),  $x_t \in \mathbb{R}^m$  τα δεδομένα εισόδου διάστασης  $m$ ,  $z_t \in \mathbb{R}^d$  τα δεδομένα εξόδου στο χρονικό βήμα  $t$ ,  $W_i \in \mathbb{R}^{d \times d}$  και  $V_i \in \mathbb{R}^{d \times m}$  οι πίνακες με τα βάρη των προηγούμενων καταστάσεων  $h_{t-1}$  και των δεδομένων  $x_t$  αντίστοιχα,  $b_i \in \mathbb{R}^d$  ο όρος μεροληψίας. Τα  $W_o$  και  $b_o$  αναφέρονται στο στρώμα εξόδου. Οι συναρτήσεις  $\phi_1$  και  $\phi_2$  είναι οι συναρτήσεις ενεργοποίησης στα κρυφά στρώματα και στο στρώμα εξόδου αντίστοιχα (π.χ.  $\phi_1$  η σιγμοειδής και  $\phi_2$  η  $\tanh$ ). Παρατηρούμε ότι κάθε κρυφή κατάσταση  $h_t$  εξαρτάται από την προηγούμενη της καθώς και από τα δεδομένα εισόδου. Αυτή η εξάρτηση επιτρέπει να λαμβάνεται υπόψη πληροφορία από προηγούμενα δεδομένα όταν αναπροσαρμόζονται οι παράμετροι στο κελί.



Σχήμα 4.12: Κελί Elman RNN

### Μειονέκτημα ERNN

Τα κελιά ERNN υποφέρουν από το πρόβλημα εξαφάνισης ή έκρηξης παραγώγων (vanishing or exploding gradient problem) όταν τα δεδομένα είναι μεγάλης διάστασης. Αυτό συνεπάγεται ότι τα κελιά αυτά δεν είναι τόσο ικανά να διατηρούν τις μακροπρόθεσμες εξαρτήσεις που μπορεί να υπάρχουν στα δεδομένα μιας χρονοσειράς.

### 4.2.2 LSTM - Long Short-term Memory

Τα κελιά LSTM, τα οποία εισήγαγαν οι Hochreiter and Schmidhuber (1997), επιλύουν το πρόβλημα των ERNN αφού είναι ικανά να αφομοιώνουν τις μακροπρόθεσμες εξαρτήσεις των δεδομένων μας.

Η βασικότερη δομή των LSTMs είναι η κυτταρική κατάσταση ( $C_t$  - cell state) η οποία είναι υπεύθυνη για την μακροπρόθεσμη μνήμη του νευρωνικού δικτύου. Στην κατάσταση αυτή τα LSTM έχουν τη δυνατότητα να αφαιρούν ή να κρατάνε πληροφορίες. Για τις λειτουργίες αυτές υπάρχουν κάποιες άλλες δομές που ονομάζονται πύλες (gates).

Τα LSTMs αποτελούνται από 3 πύλες (gates) : εισόδου (input gate), εξόδου (output gate) και λήθης ή επιλεκτική (forget gate).

#### Βήμα 1ο

Η πύλη λήθης καθορίζει πόση από την προηγούμενη πληροφορία θα διατηρηθεί στην τρέχουσα κατάσταση και πόση από την τρέχουσα θα χρησιμοποιηθεί για το forward propagation. Τα δεδομένα αυτά εισέρχονται σε μία σιγμοειδή συνάρτηση ενεργοποίησης από την οποία λαμβάνουμε ένα διάνυσμα  $f_t$  από τιμές μεταξύ 0 και 1. Αν λάβουμε την τιμή 0, τότε διαγράφει όλη την προηγούμενη πληροφορία, ενώ αν λάβουμε την τιμή 1 την διατηρεί ολόκληρη και τη μεταφέρει στο τρέχον κελί. Αν λάβουμε κάποια τιμή ανάμεσα στο 0 – 1, σημαίνει ότι διατηρεί μόνο την ποσότητα της πληροφορίας που θεωρεί σημαντική.

#### Βήμα 2ο

Αφού έχει γίνει η επιλογή πληροφορίας, περνάμε στην πύλη εισόδου η οποία αποτελείται από δύο μέρη:

1. Εισάγουμε τις τιμές της προηγούμενης κρυφής κατάστασης  $h_t$  και τα δεδομένα  $x_t$  σε μία σιγμοειδή συνάρτηση ενεργοποίησης και αποφασίζουμε ποια βάρη θα ενημερωθούν. Λαμβάνουμε δηλαδή ένα διάνυσμα  $i_t$  με τιμές ανάμεσα στο 0 και το 1.

2. Εισάγουμε τα ίδια δεδομένα σε μία συνάρτηση υπερβολικής εφαπτομένης ( $\tanh$ ) και έτσι δημιουργούμε ένα διάνυσμα  $\tilde{C}_t$  το οποίο αποτελεί τη νέα υποψήφια κυτταρική κατάσταση (cell state). Ουσιαστικά είναι οι νέες υποψήφιες τιμές των βαρών οι οποίες βρίσκονται στο διάστημα  $[-1, 1]$ .

### Βήμα 3ο

Η νέα κυτταρική δομή  $C_t$  ορίζεται ως ένας συνδυασμός της προηγούμενης  $C_{t-1}$  αφού αυτή πολλαπλασιαστεί με το διάνυσμα  $f_t$  ώστε να διατηρηθεί η απαραίτητη πληροφορία και της υποψήφιας  $\tilde{C}_t$  πολλαπλασιασμένη με το διάνυσμα  $i_t$  ώστε να ενημερωθούν τα κατάλληλα βάρη.

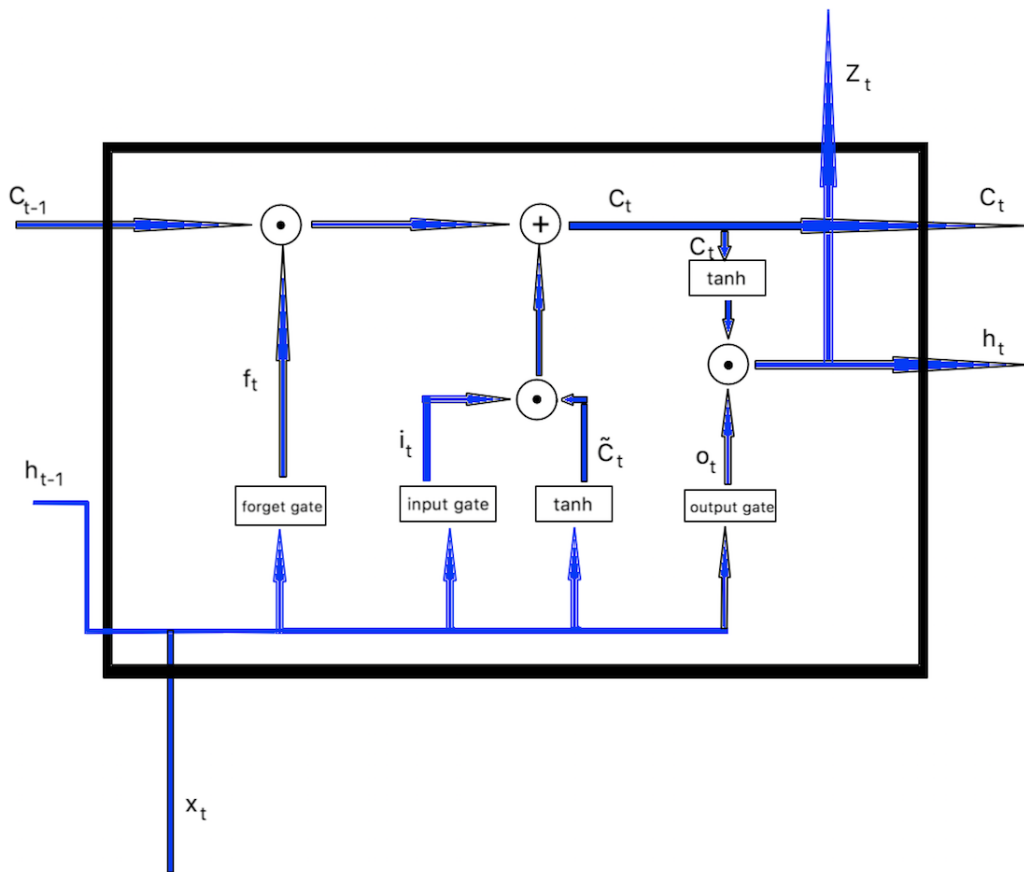
### Βήμα 4ο

Τέλος, στην πύλη εξόδου εισάγουμε τις τιμές της προηγούμενης κρυφής κατάστασης  $h_t$  και τα δεδομένα  $x_t$  σε μία σιγμοειδή συνάρτηση ενεργοποίησης και αποφασίζουμε ποια μέρη της κυτταρικής κατάστασης  $C_t$  θα εξαχθούν. Έτσι, δημιουργείται ένα διάνυσμα  $o_t$  με τιμές μεταξύ 0 και 1. Έπειτα, το πολλαπλασιάζουμε με  $\tanh(C_t)$  και λαμβάνουμε την νέα κρυφή κατάσταση  $h_t$ . Η τελική έξοδος του LSTM κελιού θα είναι η κατάσταση  $C_t$  και  $h_t$ .

Η παραπάνω διαδικασία περιγράφεται συνοπτικά από τις εξής εξισώσεις:

$$\begin{aligned} i_t &= \sigma(W_i \cdot h_{t-1} + V_i \cdot x_t + b_i) \\ o_t &= \sigma(W_o \cdot h_{t-1} + V_o \cdot x_t + b_o) \\ f_t &= \sigma(W_f \cdot h_{t-1} + V_f \cdot x_t + b_f) \\ \tilde{C}_t &= \tanh(W_c \cdot h_{t-1} + V_c \cdot x_t + b_c) \\ C_t &= i_t \circ \tilde{C}_t + f_t \circ C_{t-1} \\ h_t &= o_t \circ \tanh(C_t) \\ z_t &= h_t \end{aligned}$$

, όπου  $\circ$  το γινόμενο Hadamard.



Σχήμα 4.13: Κελί LSTM

Ένα πρόβλημα που μπορεί να προκύψει με τη χρήση LSTMs είναι η έκρηξη παραγώγων (exploding gradient problem), δηλαδή οι τιμές τους να αυξηθούν απεριόριστα πολύ. Μία λύση είναι η εφαρμογή της μεθόδου gradient clipping η οποία ελέγχει αν η παράγωγος ξεπερνάει κάποιο φράγμα  $d$  και αν ναι, την κανονικοποιεί.

### 4.2.3 GRU - Gated Recurrent Unit

Τα κελιά GRU μπορούν να θεωρηθούν ως μία παραλλαγή των LSTMs τα οποία εισήγαγαν οι Cho et al. (2014). Είναι απλούστερα και εκτελούν γρηγορότερα τους υπολογισμούς αφού διαθέτουν 2 αντί για 3 πύλες (δεν υπάρχει η πύλη εξόδου) και μόνο μία κρυφή κατάσταση  $h_t$  (δηλαδή δεν υπάρχει η κυτταρική κατάσταση  $C_t$ ).

#### Βήμα 1ο

Η πρώτη πύλη ονομάζεται πύλη αναβάθμισης (update gate) και καθορίζει πόση από την προηγούμενη πληροφορία θα κρατήσουμε. Έτσι, εισάγουμε τις τιμές της προηγούμενης κρυφής κατάστασης  $h_t$  και τα δεδομένα  $x_t$  σε μία σιγμοειδή συνάρτηση ενεργοποίησης και δημιουργούμε ένα διάνυσμα  $u_t$  με τιμές ανάμεσα στο 0 και το 1.

#### Βήμα 2ο

Η δεύτερη πύλη ονομάζεται πύλη επαναφοράς (reset gate) και καθορίζει πόση από την προηγούμενη πληροφορία θα διαγράψουμε. Έτσι, εισάγουμε τις τιμές της προηγούμενης κρυφής κατάστασης  $h_{t-1}$  και τα δεδομένα  $x_t$  σε μία σιγμοειδή συνάρτηση ενεργοποίησης και δημιουργούμε ένα διάνυσμα  $r_t$  με τιμές ανάμεσα στο 0 και το 1. Η διαφορά της πύλης αυτής με την πρώτη έγκειται στην χρησιμότητα της και στις τιμές των βαρών που αποθηκεύει.

#### Βήμα 3ο

Θεωρούμε  $\tilde{h}_t$  την νέα υποψήφια κρυφή κατάσταση στο χρονικό βήμα  $t$ . Αυτή υπολογίζεται εισάγοντας τις τιμές της προηγούμενης κρυφής κατάστασης  $h_{t-1}$ , πολλαπλασιασμένα με  $r_t$  ώστε να αφαιρεθεί όση πληροφορία δεν χρειάζεται πλέον, και τα δεδομένα  $x_t$  σε μία συνάρτηση υπερβολικής εφαπτομένης. Έτσι, οι τιμές που λαμβάνει το διάνυσμα  $\tilde{h}_t$  βρίσκονται στο διάστημα  $[-1, 1]$ .

#### Βήμα 4ο

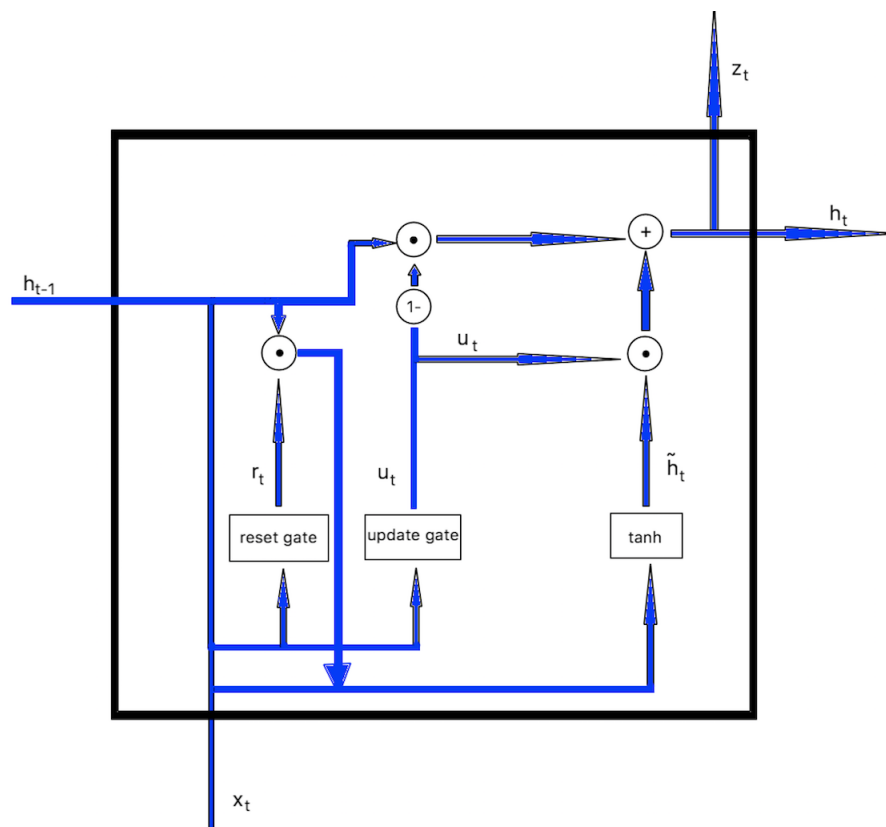
Τέλος, θέλουμε να υπολογίσουμε την τελική κατάσταση  $h_t$ . Στο σημείο αυτό χρησιμοποιείται η πύλη αναβάθμισης έτσι ώστε να κρατήσουμε τη σημαντική πληροφορία από το τρέχον και τα προηγούμενα κελιά.



Η παραπάνω διαδικασία περιγράφεται συνοπτικά από τις εξής εξισώσεις:

$$\begin{aligned}
 u_t &= \sigma(W_u \cdot h_{t-1} + V_u \cdot x_t + b_u) \\
 r_t &= \sigma(W_r \cdot h_{t-1} + V_r \cdot x_t + b_r) \\
 \tilde{h}_t &= \tanh(W_h \cdot r_t \cdot h_{t-1} + V_h \cdot x_t + b_h) \\
 h_t &= u_t \circ \tilde{h}_t + (1 - u_t) \circ h_{t-1} \\
 z_t &= h_t
 \end{aligned}$$

, όπου  $\circ$  το γινόμενο Hadamard.



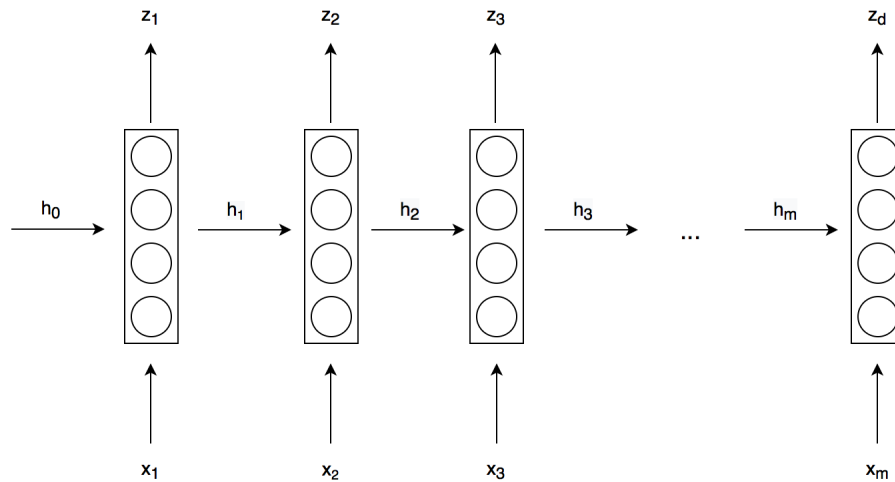
Σχήμα 4.14: Κελί GRU

**Σημείωση:** Ως συναρτήσεις ενεργοποίησης στα διάφορα κελιά μπορούν να χρησιμοποιηθούν οποιεσδήποτε ικανοποιούν τη λειτουργία τους.

Ένα ολοκληρωμένο RNN είναι ουσιαστικά μία ακολουθία κελιών. Συνηθέστερη επιλογή είναι τα κελιά LSTM, τα οποία παρ' όλο που χρειάζονται περισσότερο χρόνο εκπαίδευσης από τα GRU, φαίνεται να έχουν καλύτερα αποτελέσματα τις περισσότερες φορές.

#### 4.2.4 Forward propagation

Έστω  $m$  η διάσταση των δεδομένων εισόδου και  $d$  η διάσταση των δεδομένων εξόδου. Στην παρούσα εργασία θεωρούμε  $m = d$ . Οι παραπάνω διαδικασίες που περιγράφηκαν για κάθε είδους κελιού, αποτελούν την μέθοδο Forward propagation του δικτύου. Έτσι, ενώνοντας πολλά κελιά μεταξύ τους, σχηματίζεται ένα RNN. Ως  $h_0$  αρχικοποιείται να είναι το μηδενικό ή ένα κατάλληλο τυχαίο διάνυσμα.



Σχήμα 4.15: Forward Propagation

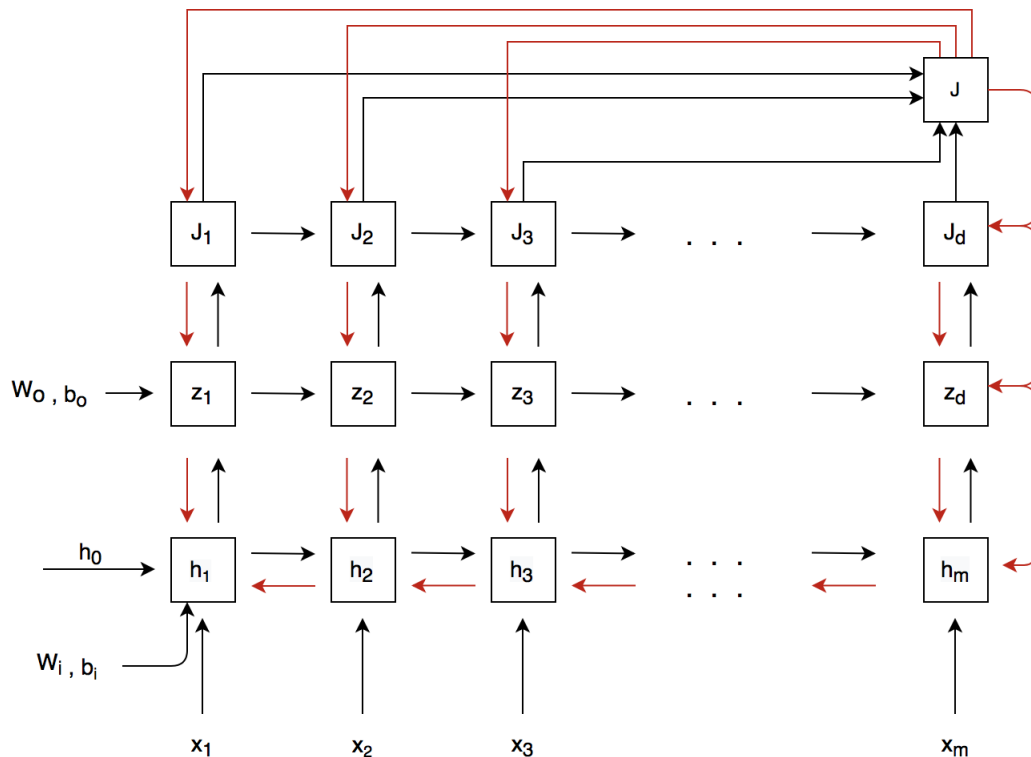
Η βασική διαφορά από τα απλά νευρωνικά είναι ότι πλέον εκτός από τα δεδομένα εισόδου  $x_t$ , υπάρχουν και οι κρυφές καταστάσεις  $h_t$  που συνδέουν τα διάφορα κελιά μεταξύ τους.

#### 4.2.5 Backward propagation (through time)

Όπως στα απλά NN, έτσι και στα RNN απαιτείται η ενημέρωση των παραμέτρων μέσω του Backward propagation. Οι εξισώσεις είναι αρκετά περίπλοκες και δεν αναφέρονται στην παρούσα εργασία.

Η λογική που ακολουθείται είναι η ίδια με των απλών NN με τη διαφορά ότι τώρα έχουμε και την κρυφή κατάσταση  $h_t$ . Με απλά λόγια, πηγαίνοντας προς τα πίσω, υπολογίζοντας παραγώγους, ενημερώνουμε τις τιμές των παραμέτρων των  $z$  ξεκινώντας από το  $z_1$  μέχρι το  $z_d$  και έπειτα των  $h$ , ξεκινώντας από το  $h_m$  μέχρι το  $h_1$ . Για την ενημέρωση, χρησιμοποιείται κάποιος αλγόριθμος βελτιστοποίησης.

Μπορούμε να αναπαραστήσουμε την διαδικασία αυτή γραφικά.



Σχήμα 4.16: Backward Propagation through time

#### 4.2.6 Bi-Directional RNN - BRNN

Η διαφορά των BRNN από τα απλά RNN είναι ότι ενώ τα δεύτερα επεξεργάζονται πληροφορίες μόνο από προηγούμενα δεδομένα, τα πρώτα είναι ικανά να επεξεργάζονται ταυτόχρονα και πληροφορίες από επόμενα δεδομένα της χρονοσειράς. Με αυτόν τον τρόπο, αυξάνεται η συνολική πληροφορία που μπορεί να δεχτεί το δίκτυο και άρα και η απόδοση του.

Ο τρόπος που λειτουργεί ένα κελί BRNN είναι ο εξής:

Προστίθεται στη δομή των απλών κελιών RNN ένα επίπεδο όπου ξεκινάει τροφοδοτούμενο με την τελευταία παρατήρηση της χρονοσειράς. Έτσι, μπορούμε να ορίσουμε ως  $h'_t$  τις κρυφές καταστάσεις που προκύπτουν από την αντίθετη κατεύθυνση. Ουσιαστικά είναι σα να έχουμε δύο RNN όπου το ένα δέχεται αντίστροφα τις παρατηρήσεις.

Η τιμή που προβλέπει το δίκτυο δίνεται από:

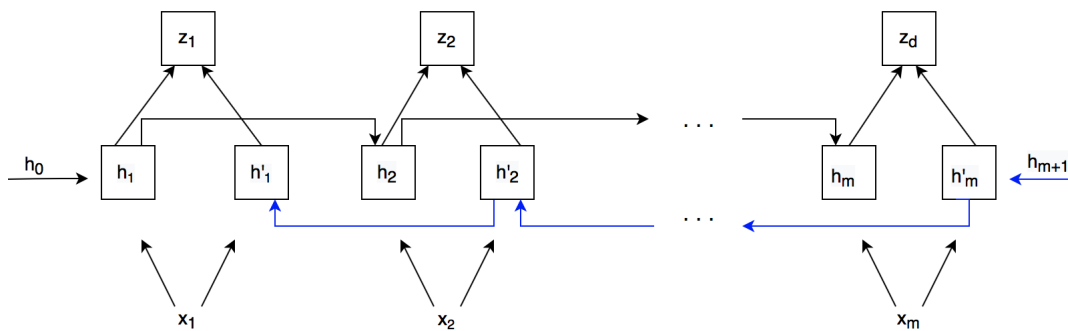
$$z_t = \phi_2(W_z \cdot h_t + V_z \cdot h'_t + b_z)$$

, όπου  $\phi_2$  μια συνάρτηση ενεργοποίησης.

Οι κρυφές καταστάσεις υπολογίζονται όμοια με το ERNN:

$$\begin{aligned} h_t &= \phi_1(W_i \cdot h_{t-1} + V_i \cdot x_t + b_i) \\ h'_t &= \phi_1(W'_i \cdot h'_{t+1} + V'_i \cdot x_t + b'_i) \end{aligned}$$

Παρακάτω παρουσιάζεται γραφικά η δομή ενός BRNN. Ως  $h_0$  και  $h_{m+1}$  αρχικοποιούνται να είναι το μηδενικό ή κάποιο άλλο κατάλληλο τυχαίο διάνυσμα.



Σχήμα 4.17: Δομή ενός BRNN

Η υλοποίηση ενός BRNN μπορεί να γίνει φυσικά επιλέγοντας είτε LSTMs είτε GRUs, εκτός των απλών ERNN κελιών. Συνήθως χρησιμοποιούνται τα Bi-RNN με κελιά LSTM, τα οποία παρά την πολύ καλή απόδοσή τους, έχουν μεγάλο υπολογιστικό κόστος.

### 4.2.7 Deep RNN

Μέχρι τώρα έχουμε περιγράψει RNN με μόνο μία κρυφή κατάσταση  $h_t$ . Στα Deep RNN μπορούμε να έχουμε παραπάνω από μία. Αυτά συνήθως χρησιμοποιούνται όταν η πολύπλοκότητα του προβλήματος και των δεδομένων μας είναι μεγάλη. Άρα αλλάζει η διάσταση των  $h_t \in \mathbb{R}^d$  και γίνεται ένας πίνακας  $H_t \in \mathbb{R}^{d \times r}$ , όπου  $d$  η διάσταση των δεδομένων εξόδου (όπου εδώ ισούται με τη διάσταση των δεδομένων εισόδου  $m$ ) και  $r$  ο αριθμός των κρυφών καταστάσεων του δικτύου. Επίσης θα ορίσουμε ένα νέο δείκτη  $[l]$  ο οποίος θα υποδηλώνει το στρώμα που ανήκει η κατάσταση  $H_t$ , δηλαδή  $H_t^{[l]}$ . Θέτουμε ως  $H_t^{[0]} = x_t$  και ορίζουμε:

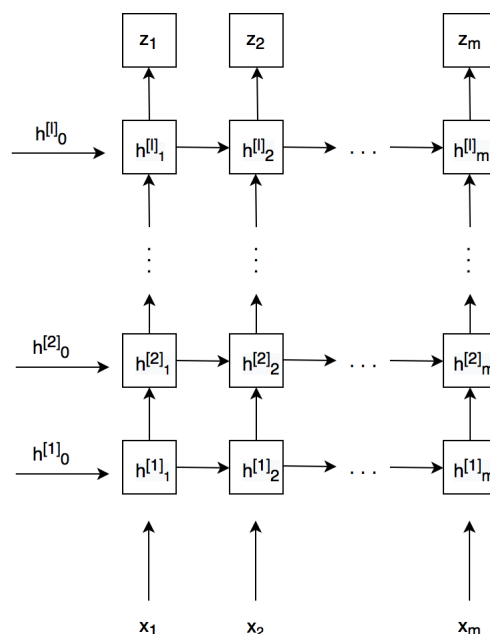
$$H_t^{[l]} = \phi_1(W_i^{[l]} \cdot H_{t-1}^{[l]} + V_i^{[l]} \cdot H_t^{[l-1]} + b_i^{[l]})$$

, όπου οι διαστάσεις των παραμέτρων του στρώματος  $l$  είναι:  $W_i^{[l]}, V_i^{[l]} \in \mathbb{R}^{r \times r}$  και  $b_i^{[l]} \in \mathbb{R}^{1 \times r}$ .

Η έξοδος κάθε στρώματος υπολογίζεται κανονικά όπως πριν.

$$Z_t^{[l]} = \phi_2(W_z^{[l]} \cdot H_{t-1}^{[l]} + V_z^{[l]} \cdot H_t^{[l-1]} + b_z^{[l]})$$

Παρακάτω παρουσιάζεται γραφικά η δομή ενός Deep RNN.



Σχήμα 4.18: Δομή ενός Deep RNN

## 5 Περιγραφή πειράματος και παρουσίαση αποτελεσμάτων

Σκοπός του πειράματος είναι να δούμε κατά πόσο ένα αναδρομικό νευρωνικό δίκτυο είναι σε θέση να προβλέψει την επόμενη τιμή της χρονοσειράς, υπολογίζοντας έτσι τα υπόλοιπα (σφάλματα πρόβλεψης). Δηλαδή θα αντικαταστήσουμε τα μοντέλα ARIMA και ETS από ένα RNN. Αφού έχουν υπολογιστεί τα υπόλοιπα και πληρούν τις προϋποθέσεις κανονικότητας και ανεξαρτησίας, θα εφαρμόσουμε σε αυτά τα κλασικά διαγράμματα EWMA και CUSUM. Θα συγκρίνουμε την απόδοση αυτών με τα διαγράμματα που περιγράψαμε στο κεφάλαιο 2, με βάση το μέσο μήκος ροής *ARL*.

### Γενική δομή και λειτουργία του αναδρομικού νευρωνικού δικτύου

Προεπεξεργασία των δεδομένων εισόδου και βασική δομή:

1. Γίνεται τυποποίηση των τιμών της  $X_t$  αφαιρώντας την τιμή στόχο  $\mu_0$  και διαιρώντας με την τυπική απόκλιση της  $\sigma$ . Ο λόγος για τον οποίο γίνεται αυτό είναι διότι αν υπάρχουν τιμές στα δεδομένα που κυμαίνονται σε μεγάλη κλίμακα, το νευρωνικό θα θεωρήσει λανθασμένα ότι αυτές είναι πιο σημαντικές και θα τους προσδώσει μεγαλύτερα βάρη. Μία συνέπεια αυτού μπορεί να είναι η αργή εκπαίδευση του δικτύου. Ως τιμή στόχο  $\mu_0$  ορίσαμε να είναι η μέση τιμή της  $X_t$  όταν αυτή δεν έχει μετατοπιστεί.
2. Το νευρωνικό δίκτυο προβλέπει τις επόμενες τιμές της χρονοσειράς βασιζόμενο στις lag τιμές της. Οι lag τιμές μιας χρονοσειράς εξηγούνται στο κεφάλαιο 2. Αν ορίσουμε το lag=12 σημαίνει ότι το δίκτυο χρησιμοποιεί τις 12 προηγούμενες τιμές συνεχώς για να προβλέψει τις επόμενες. Το πλήθος των τιμών που θέλουμε να προβλέψουμε (predictions) ορίζουμε να είναι ίσο με τα lags. Τα predictions μπορούν να οριστούν διαφορετικά από τα lags αλλά σίγουρα όχι περισσότερα σε πλήθος.
3. Το σύνολο δεδομένων χωρίζεται σε σύνολο εκπαίδευσης και σύνολο αξιολόγησης. Το σύνολο εκπαίδευσης μας αποτελείται από τον αριθμό των παρατηρήσεων μειωμένος κατά τα lags και τα predictions και αυξημένος κατά ένα. Το σύνολο αξιολόγησης ορίζεται να είναι οι τελευταίες lag το πλήθος παρατηρήσεις της χρονοσειράς.

4. Η μεταβλητή απόκρισης  $y$  του δικτύου ορίζεται να είναι οι επόμενες τιμές που πρέπει να προβλέψει το νευρωνικό δίκτυο. Ξεκινάει από την  $\text{lag}+1$  παρατήρηση καθώς οι πρώτες  $\text{lag}$  το πλήθος έχουν χρησιμοποιηθεί για την πρώτη πρόβλεψη. Οι τιμές της  $y$  λαμβάνονται από το σύνολο εκπαίδευσης.
5. Το σύνολο αξιολόγησης χρησιμοποιείται στο τέλος για να ελέγξουμε την απόκριση του εκπαιδευμένου νευρωνικού δικτύου.

### Υπολογισμός σφαλμάτων πρόβλεψης (υπόλοιπα - residuals)

Μετά την εκπαίδευση του νευρωνικού δικτύου, ανασκευάζουμε τις προβλεπόμενες τιμές (fitted values) πολλαπλασιάζοντας με την τυπική απόκλιση  $\sigma$  και προσθέτοντας την μέση τιμή στόχο  $\mu_0$ . Ο λόγος που γίνεται το βήμα αυτό είναι για να πάρουμε ξανά τα δεδομένα στην κλίμακα που δόθηκαν. Τέλος, για να υπολογίσουμε τα σφάλματα πρόβλεψης παίρνουμε την διαφορά αυτών από τις πραγματικές τιμές.

### Monte Carlo Simulations

Προσομοιώσαμε 1000 χρονοσειρές που ακολουθούν το μοντέλο που εξετάζουμε κάθε φορά (με 1000 παρατηρήσεις  $\eta$  κάθε μία) για κάθε μετατόπιση  $\alpha$  στο μέσο της, όπου  $\alpha \in \{0, 0.5, 1.0, 1.5, \dots, 3.0\}$ .

#### 1. Τροποποιημένο EWMA - (εν συντομία ModEWMA)

Για κάθε μοντέλο, εξετάζουμε διάφορους συνδυασμούς των  $\lambda$  και  $L$  και βρίσκουμε το πρώτο σημείο που βγαίνει εκτός ορίων ενώ η διεργασία μας είναι εντός ελέγχου ( $\alpha = 0$ ). Τα σημεία αυτά είναι τα εντός ελέγχου μήκη ροής ( $RL_0$ ) του διαγράμματος για τους διάφορους συνδυασμούς ( $\lambda, L$ ). Τα  $\lambda$  που δοκιμάστηκαν ανήκουν στο διάστημα  $[0.05, 0.95]$  με βήμα 0.05. Ο τρόπος με τον οποίο υπολογίστηκαν τα αντίστοιχα  $L$  που δίνουν μέσο μήκος ροής  $ARL_0 \approx 370$  είναι ο εξής:

Αρχικά αναζητούμε  $L$  στο διάστημα  $[1, 3]$  με βήμα 0.05. Βρίσκουμε τις τιμές των  $RL_0$  για όλους τους συνδυασμούς ( $\lambda, L$ ) που έχουμε ως τώρα. Επαναλαμβάνοντας 1000 φορές, παίρνουμε τον μέσο όρο των τιμών αυτών, το λεγόμενο εντός ελέγχου μέσο μήκος ροής  $ARL_0$  που περιγράψαμε στο 2ο κεφάλαιο. Επιλέγουμε τους συνδυασμούς ( $\lambda, L$ ) που μας δίνουν  $300 \leq ARL_0 \leq 450$  και επαναλαμβάνουμε την παραπάνω διαδικασία με το

$L$  να ανήκει στο διάστημα  $(L_1, L_2)$  με βήμα 0.001, όπου  $L_1$  το σημείο όπου έχουμε  $ARL_0 = 300$  και  $L_2$  το σημείο όπου  $ARL_0 = 450$ . Τελικά κρατάμε εκείνα τα ζεύγη  $(\lambda, L)$  που δίνουν  $ARL_0 \approx 370$ . Ο λόγος που επαναλαμβάνεται η διαδικασία με μικρότερο βήμα στο  $L$  είναι για να έχουμε όσο μεγαλύτερη ακρίβεια γίνεται στις τελικές επιλογές των  $(\lambda, L)$ .

Αφού έχει γίνει η τελική επιλογή των  $(\lambda, L)$ , για κάθε μία μετατόπιση  $a$  υπολογίζουμε το πρώτο σημείο όπου η διεργασία βγαίνει εκτός ορίων. Αυτά είναι τα λεγόμενα εκτός ελέγχου μήκη ροής  $RL_1$ . Επαναλαμβάνουμε την παραπάνω διαδικασία 1000 φορές και παίρνοντας τους μέσους όρους για τα συγκεκριμένα  $(\lambda, L)$ , λαμβάνουμε τις τιμές των εκτός ελέγχου μηκών ροής  $ARL_1$ . Οι τιμές αυτών θέλουμε να είναι μικρές, ώστε το διάγραμμα να αντιλαμβάνεται όσο πιο γρήγορα γίνεται την μετατόπιση που έχει προκύψει στο μέσο.

## 2. Τροποποιημένο CUSUM - (εν συντομία ModCUSUM)

Όμοια με το EWMA, οι υπολογισμοί των βέλτιστων ζευγών  $(h, k)$ , έγιναν για  $ARL_0 = 370$ . Η διαδικασία που ακολουθήθηκε είναι εντελώς παρόμοια με το EWMA. Το διάστημα από το οποίο επιλέχθηκαν τα  $k$  είναι το  $[0.25, 1.5]$  με βήμα 0.25 και τα  $h$  αρχικά από το  $[1, 15]$  με βήμα 0.05. Αφού υπολογίσαμε τα  $ARL_0$  προσομοιώνοντας 1000 χρονοσειρές για όλους τους συνδυασμούς των  $(h, k)$ , κρατήσαμε για κάθε  $k$  τα  $h$  εκείνα που έδιναν  $300 \leq ARL_0 \leq 450$ . Όμοια με το EWMA, επαναλάβαμε την παραπάνω διαδικασία για τα νέα διαστήματα των  $h$  με βήμα 0.001 και κρατήσαμε τα ζεύγη  $(h, k)$  που έδιναν  $ARL_0 \approx 370$ .

Για τον υπολογισμό των  $ARL_1$ , προσομοιώσαμε ξανά 1000 χρονοσειρές για κάθε μετατόπιση  $a$  και επιλέξαμε από τα τελικά ζεύγη  $(h, k)$  εκείνο που έδινε κάθε φορά την μικρότερη τιμή.

## 3. EWMA και CUSUM στα υπόλοιπα των μοντέλων ARIMA και ETS - (εν συντομία ResEWMA και ResCUSUM αντίστοιχα)

Στην περίπτωση των διαγραμμάτων υπολοίπων, υπολογίζουμε τα υπόλοιπα για κάθε μετατοπισμένη διαδικασία  $X_t$  που ακολουθεί το μοντέλο που εξετάζουμε. Αφού τα κανονικοποιήσουμε όπως περιγράψαμε στο κεφάλαιο 2, ελέγχουμε αν αυτά είναι ανεξάρτητα και αν ακολουθούν τελικά την κανονική κατανομή. Στην περίπτωση που ισχύουν οι προϋποθέσεις αυτές, είμαστε σε θέση να εφαρμόσουμε σε αυτά τα κλασικά διαγράμματα



## EWMA και CUSUM.

Τα υπόλοιπα ικανοποιούσαν τις προϋποθέσεις κανονικότητας και ανεξαρτησίας και οι βέλτιστες τιμές των παραμέτρων για  $ARL_0 = 370$  υπάρχουν ήδη στη βιβλιογραφία.

Για το EWMA, επιλέξαμε 5 ζεύγη  $(\lambda, L)$  με βάση τη βιβλιογραφία:  $(0.1, 2.698)$ ,  $(0.2, 2.856)$ ,  $(0.3, 2.929)$ ,  $(0.4, 2.956)$ ,  $(0.5, 2.977)$ . Για τα ζεύγη αυτά προσομοιώσαμε 1000 χρονοσειρές ώστε να επιβεβαιώσουμε ότι δίνουν  $ARL_0 \approx 370$ . Έπειτα για κάθε μετατόπιση  $\alpha$ , βρήκαμε τα  $ARL_1$  προσομοιώνοντας ξανά 1000 χρονοσειρές για κάθε ζεύγος και επιλέξαμε εκείνο που έδινε την μικρότερη τιμή.

Για το CUSUM, επιλέξαμε 6 ζεύγη  $(h, k)$  με βάση τη βιβλιογραφία:  $(8.01, 0.25)$ ,  $(4.77, 0.5)$ ,  $(3.34, 0.75)$ ,  $(2.52, 1)$ ,  $(1.99, 1.25)$ ,  $(1.61, 1.50)$ . Για τα ζεύγη αυτά προσομοιώσαμε 1000 χρονοσειρές ώστε να επιβεβαιώσουμε ότι δίνουν  $ARL_0 \approx 370$ . Έπειτα για κάθε μετατόπιση  $\alpha$ , βρήκαμε τα  $ARL_1$  προσομοιώνοντας ξανά 1000 (μετατοπισμένες) χρονοσειρές για κάθε ζεύγος και επιλέξαμε εκείνο που έδινε την μικρότερη τιμή.

Οι τιμές στόχοι, των υπολοίπων, ορίζουμε να είναι η μέση τιμή και η διασπορά εκείνων που λαμβάνουμε όταν εφαρμόζουμε το μοντέλο στην μη μετατοπισμένη χρονοσειρά.

#### 4. EWMA και CUSUM στα υπόλοιπα του νευρωνικού δικτύου - (εν συντομία ResEWMArnn και ResCUSUMrnn)

##### ResEWMArnn

Για το κλασικό EWMA, αρχικά εισάγουμε τις προσομοιωμένες χρονοσειρές χωρίς μετατόπιση εξετάζοντας για ποια ζεύγη  $(\lambda, L)$  έχουμε  $ARL_0 \approx 370$ , όπου  $\lambda \in [0.05, 0.95]$  και  $L \in [1, 3]$ . Όμοια με πριν, πρώτα εξετάζονται τα  $L$  με βήμα 0.05 και έπειτα αφού μειώσουμε το διάστημα, με βήμα 0.001. Για τον υπολογισμό των  $ARL_1$ , μετατοπίζουμε τις 1000 προσομοιωμένες χρονοσειρές κατά  $\alpha\sigma_0$ , όπου  $\alpha \in \{0.5, 1, 1.5, 2, 2.5, 3\}$  και  $\sigma_0 = \sqrt{Var(Y_t)}$  και αφού βρούμε τα υπόλοιπα μέσω του εκπαιδευμένου RNN, ελέγχουμε για ποια τελικά ζεύγη  $(\lambda, L)$  επιτυγχάνεται μικρότερη τιμή.

##### ResCUSUMrnn

Για το κλασικό CUSUM, αρχικά εισάγουμε τις προσομοιωμένες χρονοσειρές χωρίς μετατόπιση εξετάζοντας για ποια ζεύγη  $(h, k)$  έχουμε  $ARL_0 \approx 370$ . Τα διαστήματα που επιλέχθηκε το  $h$ , εξαρτώνται από τη συσχέτιση που μελετάμε. Το εύρος τους είναι  $[1, 15]$  και  $k \in [0.25, 1.5]$  με βήμα 0.05. Όμοια με πριν, πρώτα εξετάζονται τα  $h$

με βήμα 0.05 και έπειτα αφού μειώσουμε το διάστημα, με βήμα 0.001. Για τον υπολογισμό των  $ARL_1$ , μετατοπίζουμε τις 1000 προσομοιωμένες χρονοσειρές κατά  $\alpha\sigma_0$ , όπου  $\alpha \in \{0.5, 1, 1.5, 2, 2.5, 3\}$  και  $\sigma_0 = \sqrt{Var(Y_t)}$  και αφού βρούμε τα υπόλοιπα μέσω του εκπαιδευμένου RNN, ελέγχουμε για ποια τελικά ζεύγη  $(h, k)$  επιτυγχάνεται μικρότερη τιμή.

## 5.1 Σύγκριση για διαδικασία AR(1)

Έστω μια AR(1) διαδικασία-στόχος:

$$Y_t = \phi_1 Y_{t-1} + \epsilon_t$$

, όπου  $\epsilon_t \sim WN(0, \sigma^2)$ . Χωρίς βλάβη της γενικότητας, θεωρούμε την μέση τιμή στόχο  $\mu_0 = 0$ . Για την παρατηρούμενη χρονοσειρά  $X_t$  ισχύει:

$$X_t = Y_t + \alpha\sqrt{\gamma_0}$$

, όπου  $\alpha \in \{0, 0.5, 1.0, 1.5, \dots, 3.0\}$  και  $\gamma_0 = Var(X_t) = \frac{\sigma^2}{1 - \phi_1^2} = \sigma_0^2$ . Δηλαδή οι μετατοπίσεις  $\alpha$  της χρονοσειράς  $X_t$  που μελετάμε βρίσκονται σε κλίμακα  $\sigma_0$ . Σκοπός των διαγραμμάτων, όπως έχουμε ήδη περιγράψει, είναι να εντοπίσει όσο πιο γρήγορα γίνεται αυτή την μετατόπιση που συμβαίνει στο μέσο της διεργασίας.

Θα ελέγξουμε για συσχετίσεις  $\phi_1 \in \{-0.8, -0.2, 0.2, 0.8\}$ . Ο λόγος που επιλέξαμε τις συγκεκριμένες τιμές ήταν για να ελέγξουμε την απόδοση των διαγραμμάτων σε πολύ αρνητικές, πολύ θετικές και ενδιάμεσες συσχετίσεις.

Υλοποιήσαμε 2 αναδρομικά νευρωνικά δίκτυα, ένα δεχόμενο ως είσοδο μία διαδικασία AR(1) με συσχέτιση  $\phi_1 = 0.5$  (ας το συμβολίζουμε με posRNN) και ένα με συσχέτιση  $\phi_1 = -0.5$  (ας το συμβολίζουμε με negRNN). Μετά την εκπαίδευση τους, τα δύο νευρωνικά δίκτυα είναι σε θέση να προβλέψουν διαδικασίες με θετικές 0.8, 0.2 (posRNN) και αρνητικές  $-0.8, -0.2$  (negRNN) συσχετίσεις. Δεν χρειάζεται να υλοποιήσουμε 4 νευρωνικά (ένα για κάθε συσχέτιση) καθώς αυτό, αν εκπαιδευτεί με τις σωστές τιμές των υπερπαραμέτρων, είναι σε θέση να καλύψει ένα ευρύ φάσμα παρόμοιων δεδομένων. Ακολουθείται η βασική δομή που περιγράφηκε στην αρχή του κεφαλαίου. Παρακάτω θα παρουσιάσουμε συνοπτικά την αρχιτεκτονική που εφαρμόσαμε για το παρόν πρόβλημα.

## Αρχιτεκτονική του δικτύου

Ορισμός υπερπαραμέτρων:

- lags : 24
- predictions : 24
- Εποχές : 500
- Αριθμός κόμβων κρυφών στρωμάτων : 60
- Dropout rate : 0.2
- Συνάρτηση κόστους : Μέσο τετραγωνικό σφάλμα
- Αλγόριθμος βελτιστοποίησης : Adam

Αναλυτικότερα, το πρώτο στρώμα του δικτύου είναι ένα στρώμα αναδρομικού νευρωνικού δικτύου με χρήση κελιών LSTMs. Ο αριθμός των κόμβων σε κάθε κελί LSTM ορίζεται να είναι 60. Στα δεδομένα εξόδου του στρώματος αυτού εφαρμόζεται τεχνική Dropout με ποσοστό 0.2%.

Το δεύτερο στρώμα, αποτελεί άλλο ένα στρώμα αναδρομικού νευρωνικού δικτύου με τις ίδιες υπερπαραμέτρους. Εφαρμόζεται ξανά τεχνική Dropout με ίδιο ποσοστό.

Τέλος, το τελευταίο στρώμα, δηλαδή το στρώμα εξόδου, αποτελεί ένα απλό στρώμα νευρωνικού δικτύου όπου εφαρμόζεται η συνάρτηση απώλειας που έχουμε επιλέξει δίνοντας τελικά ως έξοδο 1 προβλεπόμενη τιμή.

Ως αλγόριθμος βελτιστοποίησης για τη συνάρτηση κόστους MSE (Μέσο τετραγωνικό σφάλμα) επιλέχθηκε ως καλύτερος ο Adam. Λόγω του τετραγωνισμού των σφαλμάτων, η MSE ποινικοποιεί τις μεγάλες αποκλίσεις μεταξύ των πραγματικών τιμών  $y$  και των προβλεπόμενων τιμών  $\hat{y}$  πιο έντονα. Για το εγχυρότητα των μοντέλων μας, θέλουμε να αποφύγουμε να αποκτήσουμε ένα μοντέλο που έχει εξαιρετική απόδοση σε ορισμένες περιπτώσεις και πολύ κακή σε άλλες. Για αυτόν τον λόγο, προτιμάται η MSE ως συνάρτηση απώλειας, καθώς προσδίδει σταθερότητα και γενίκευση του μοντέλου σε πολλές χρονοσειρές.

Μία άλλη επιλογή συνάρτησης κόστους που εξετάσαμε ήταν το μέσο απόλυτο σφάλμα (MAE). Σε αντίθεση με την MSE, η MAE είναι λιγότερο ευαίσθητη σε ακραίες τιμές και στις μεγάλες αποκλίσεις των προβλέψεων από τις πραγματικές τιμές. Με άλλα λόγια, η MAE ποινικοποιεί τα λάθη για όλες τις παρατηρήσεις  $i$  εξίσου κρίνοντας την λιγότερο κατάλληλη όταν μια πιθανή ακραία τιμή έχει μεγάλη αρνητική επίδραση για την πρακτική χρήση του μοντέλου.

Οι εποχές επιλέχθηκαν έτσι ώστε να αποφευχθεί το overfitting και ταυτόχρονα επειδή παρατηρήθηκε ότι η συνάρτηση κόστους μετά τις 500 εποχές δεν παρουσίαζε κάποια βελτίωση.

Αφού εκπαιδευτεί το δίκτυο, προσομοιώνουμε 1000 χρονοσειρές από το μοντέλο  $AR(1)$  με την συσχέτιση που μας ενδιαφέρει κάθε φορά και κάνοντας τις προβλέψεις μέσω του RNN για κάθε μία από αυτές, λαμβάνουμε τα υπόλοιπα. Παρατηρήθηκε ότι για μεγάλες συσχετίσεις (0.8,-0.8) τα υπόλοιπα, παρόλο που ακολουθούσαν την κανονική κατανομή, δεν ήταν ανεξάρτητα, οπότε εφαρμόσαμε σε αυτά τα τροποποιημένα διαγράμματα ελέγχου.

Ακολουθείται ακριβώς αντίστοιχη διαδικασία εύρεσης βέλτιστων τιμών των παραμέτρων για τα ResModEWMArnn και ResModCUSUMrnn.

Πίνακας 5.1: Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη  $ARL_0 \approx 370$ 

ModEWMA						
$ARL_0 \approx 370$	Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	0.5	1	1.5	2	2.5	3
0.8	$\lambda=0.05$ L=2.982	$\lambda=0.05$ L=2.982	$\lambda=0.05$ L=2.982	$\lambda=0.850$ L=2.229	$\lambda=0.95$ L=2.112	$\lambda=0.95$ L=2.112
0.2	$\lambda=0.05$ L=1.938	$\lambda=0.1$ L=2.074	$\lambda=0.3$ L=2.194	$\lambda=0.35$ L=2.214	$\lambda=0.45$ L=2.219	$\lambda=0.7$ L=2.180
-0.2	$\lambda=0.1$ L=1.778	$\lambda=0.15$ L=1.843	$\lambda=0.35$ L=1.980	$\lambda=0.35$ L=1.980	$\lambda=0.5$ L=2.013	$\lambda=0.5$ L=2.013
-0.8	$\lambda=0.2$ L=1.028	$\lambda=0.2$ L=1.028	$\lambda=0.3$ L=1.172	$\lambda=0.35$ L=1.241	$\lambda=0.35$ L=1.241	$\lambda=0.3$ L=1.172

Πίνακας 5.2: Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη  $ARL_0 \approx 370$ 

ModCUSUM						
$ARL_0 \approx 370$	Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	0.5	1	1.5	2	2.5	3
0.8	k=0.25 h=3.354	k=0.50 h=2.086	k=0.75 h=1.241	k=1.00 h=1.182	k=1.25 h=1.115	k=1.5 h=1.093
0.2	k=0.25 h=6.495	k=0.50 h=4.637	k=0.75 h=3.254	k=1.00 h=2.213	k=1.25 h=2.034	k=1.5 h=1.599
-0.2	k=0.25 h=6.420	k=0.50 h=4.239	k=0.75 h=3.106	k=1.00 h=2.187	k=1.25 h=1.927	k=1.5 h=1.528
-0.8	k=0.25 h=2.053	k=0.50 h=1.782	k=0.75 h=1.213	k=1.00 h=1.121	k=1.25 h=1.086	k=1.5 h=1.054

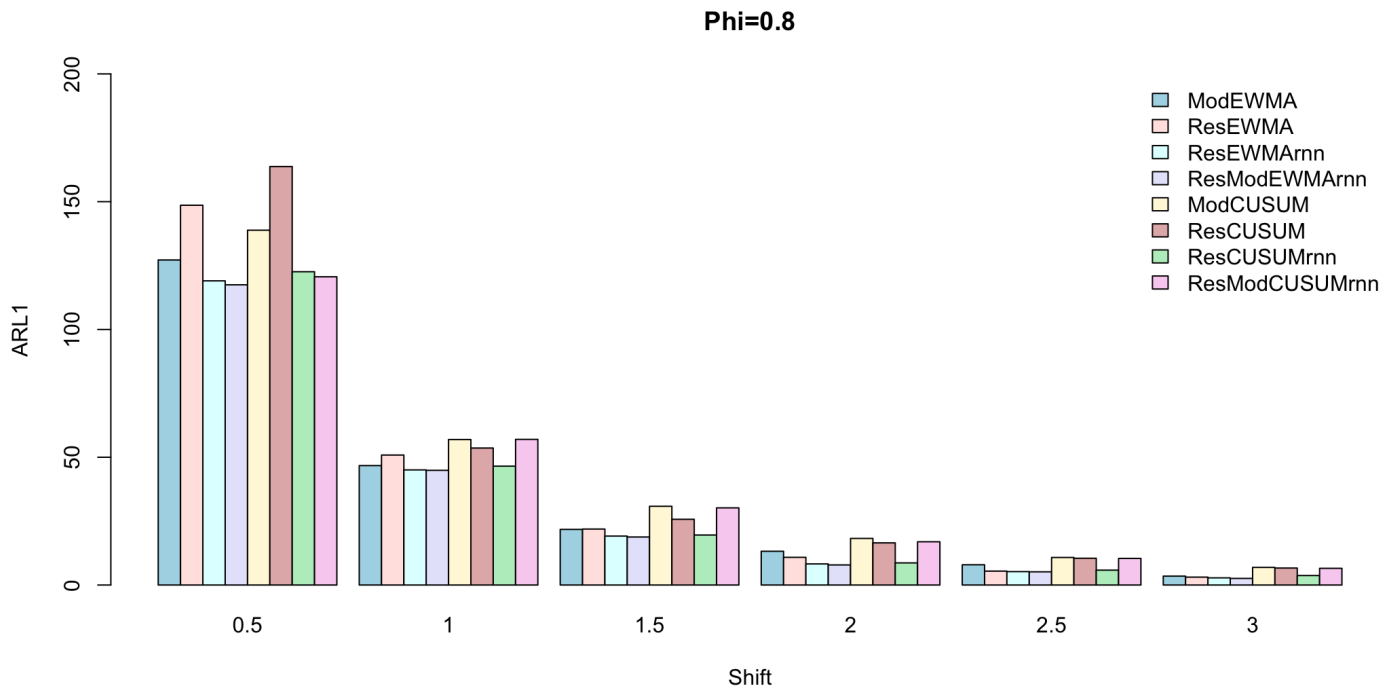
Πίνακας 5.3: Σύγκριση διαγραμμάτων με βάση  $ARL_1$  για μία AR(1) διαδικασία, όταν  $ARL_0 = 370$

AR(1)		Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	Μέθοδος	0.5	1	1.5	2	2.5	3
0.8	ModEWMA	127.190	46.725	21.760	13.220	7.925	3.485
	ModCUSUM	138.845	56.935	30.810	18.240	10.785	6.890
	ResEWMA	148.610	50.860	21.900	10.840	5.440	3.085
	ResCUSUM	163.760	53.605	25.730	16.495	10.465	6.650
	ResEWMArnn	119.020	45.040	19.155	8.260	5.230	2.800
	ResCUSUMrnn	122.595	46.515	19.565	8.655	5.830	3.705
	ResModEWMArnn	<b>117.470</b>	<b>44.875</b>	<b>18.790</b>	<b>7.860</b>	<b>5.160</b>	<b>2.590</b>
	ResModCUSUMrnn	120.610	57.000	30.190	16.920	10.405	6.545
0.2	ModEWMA	34.405	12.820	6.905	4.185	2.975	2.065
	ModCUSUM	40.395	13.465	6.870	4.240	2.870	1.860
	ResEWMA	34.890	9.695	4.720	2.870	1.930	1.490
	ResCUSUM	41.065	13.335	6.830	4.210	2.840	1.845
	ResEWMArnn	<b>29.795</b>	<b>7.320</b>	<b>3.765</b>	<b>2.445</b>	<b>1.785</b>	<b>1.465</b>
	ResCUSUMrnn	35.130	11.030	5.180	3.165	2.235	1.745
-0.2	ModEWMA	19.875	7.660	3.865	2.760	2.025	1.580
	ModCUSUM	21.510	7.940	4.095	2.750	2.105	1.650
	ResEWMA	18.610	5.810	3.075	2.170	1.690	1.400
	ResCUSUM	21.300	6.935	4.075	2.665	2.005	1.600
	ResEWMArnn	<b>16.545</b>	<b>4.470</b>	<b>2.410</b>	<b>1.635</b>	<b>1.335</b>	<b>1.220</b>
	ResCUSUMrnn	25.205	6.790	3.550	2.305	1.795	1.570
-0.8	ModEWMA	6.896	2.980	1.795	1.540	1.355	1.195
	ModCUSUM	5.875	2.615	2.030	1.975	1.890	1.800
	ResEWMA	5.120	2.410	1.960	1.760	1.610	1.390
	ResCUSUM	5.870	2.590	1.985	1.870	1.740	1.580
	ResEWMArnn	<b>5.090</b>	<b>1.845</b>	<b>1.155</b>	<b>1.005</b>	<b>1.000</b>	<b>1.000</b>
	ResCUSUMrnn	9.950	3.980	1.995	1.410	1.150	1.060
	ResModEWMArnn	9.175	3.440	1.750	1.955	1.080	1.010
	ResModCUSUMrnn	8.475	2.900	2.000	1.940	1.705	1.365

### 5.1.1 Σχολιασμός αποτελεσμάτων

Αρχικά είναι φανερό από τον παραπάνω πίνακα ότι όσο πιο αρνητική είναι η συσχέτιση τόσο καλύτερη απόδοση έχουν τα διαγράμματα.

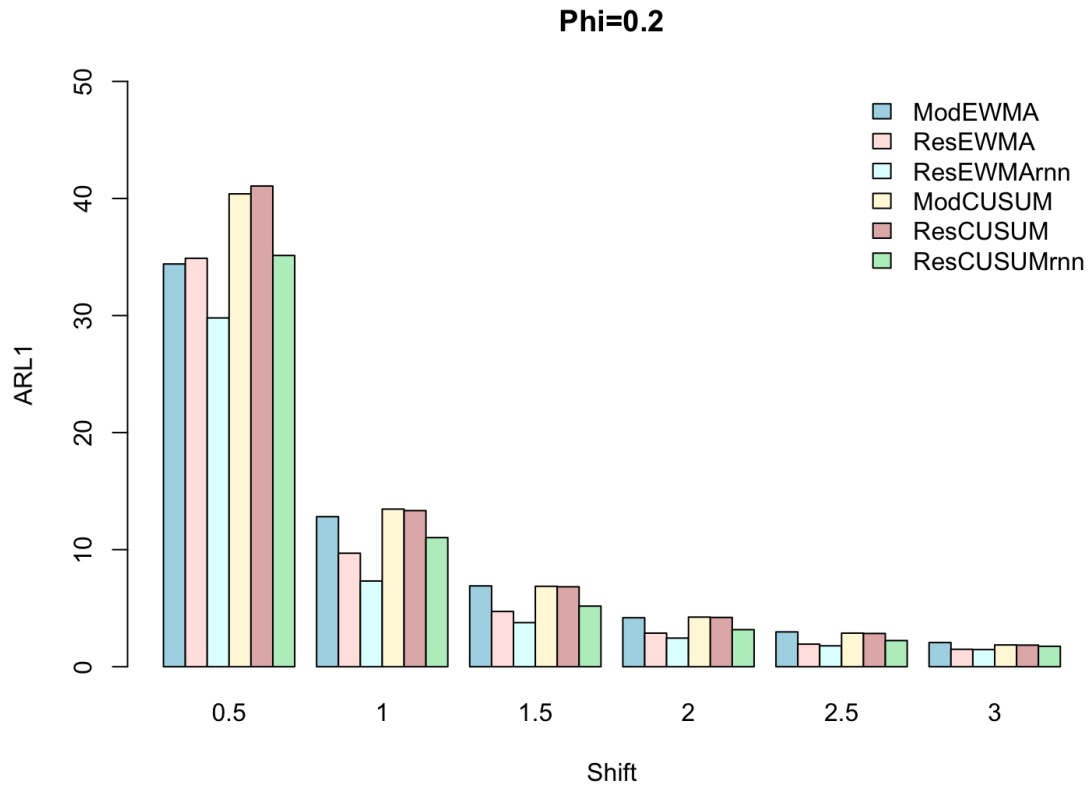
Παρακάτω βλέπουμε για κάθε συσχέτιση και κάθε μετατόπιση ξεχωριστά τις τιμές των  $ARL_1$  στο ίδιο διάγραμμα.



Σχήμα 5.1:  $ARL_1$  για συσχέτιση  $\phi_1 = 0.8$  στα διάφορα διαγράμματα

Παρατηρούμε ότι για συσχέτιση  $\phi_1 = 0.8$  τα διαγράμματα EWMA υπερτερούν σε σύγκριση με τα αντίστοιχα CUSUM. Πιο συγκεκριμένα, για μικρή μετατόπιση, 0.5, φαίνεται ότι τα τροποποιημένα διαγράμματα ModEWMA, ModCUSUM έχουν καλύτερη απόδοση από τα απλά διαγράμματα υπολοίπων. Τα διαγράμματα υπολοίπων από το RNN όμως, και μάλιστα τα τροποποιημένα, φαίνεται να υπερτερούν από όλα τα υπόλοιπα διαγράμματα και για τα δύο είδη, EWMA και CUSUM.

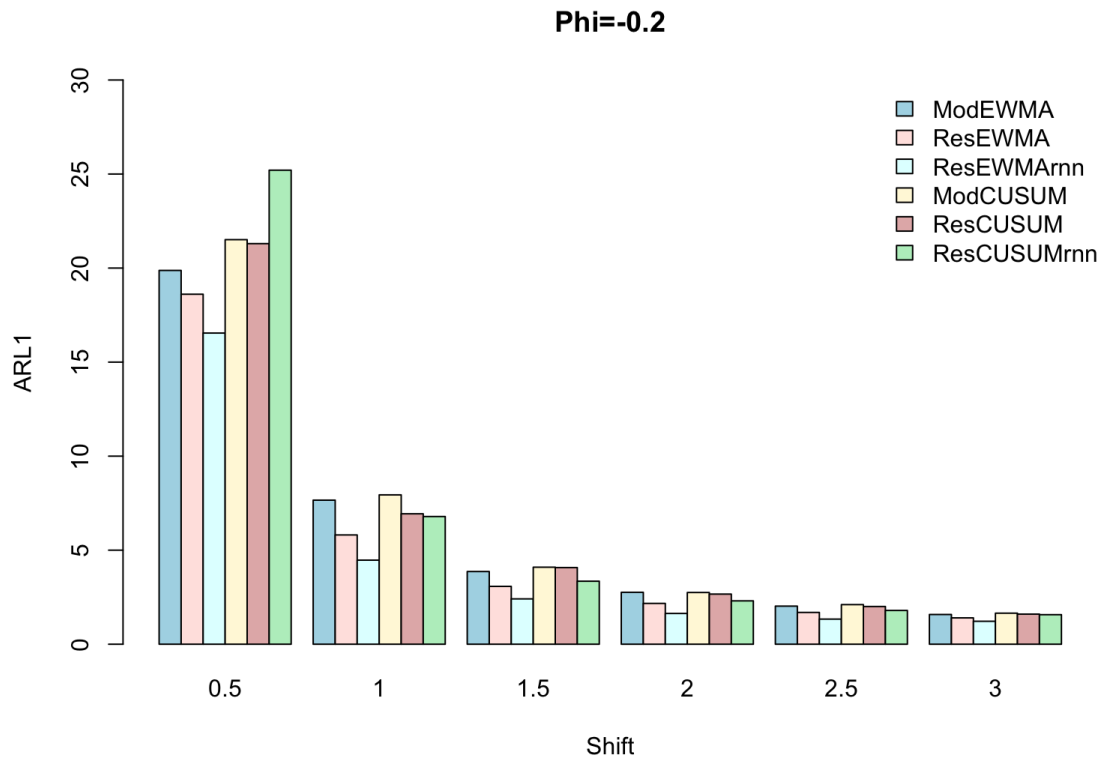
Όσο αυξάνεται η τιμή της μετατόπισης, φαίνεται ότι τα απλά διαγράμματα υπολοίπων αποδίδουν καλύτερα από τα τροποποιημένα. Για το EWMA, τα διαγράμματα υπολοίπων του RNN έχουν την καλύτερη απόδοση για όλες τις μετατοπίσεις. Για το CUSUM, φαίνεται να υπερτερεί το ResCUSUMrnn, χωρίς όμως να ξεπερνάει το EWMA.



Σχήμα 5.2:  $ARL_1$  για συσχέτιση  $\phi_1 = 0.2$  στα διάφορα διαγράμματα

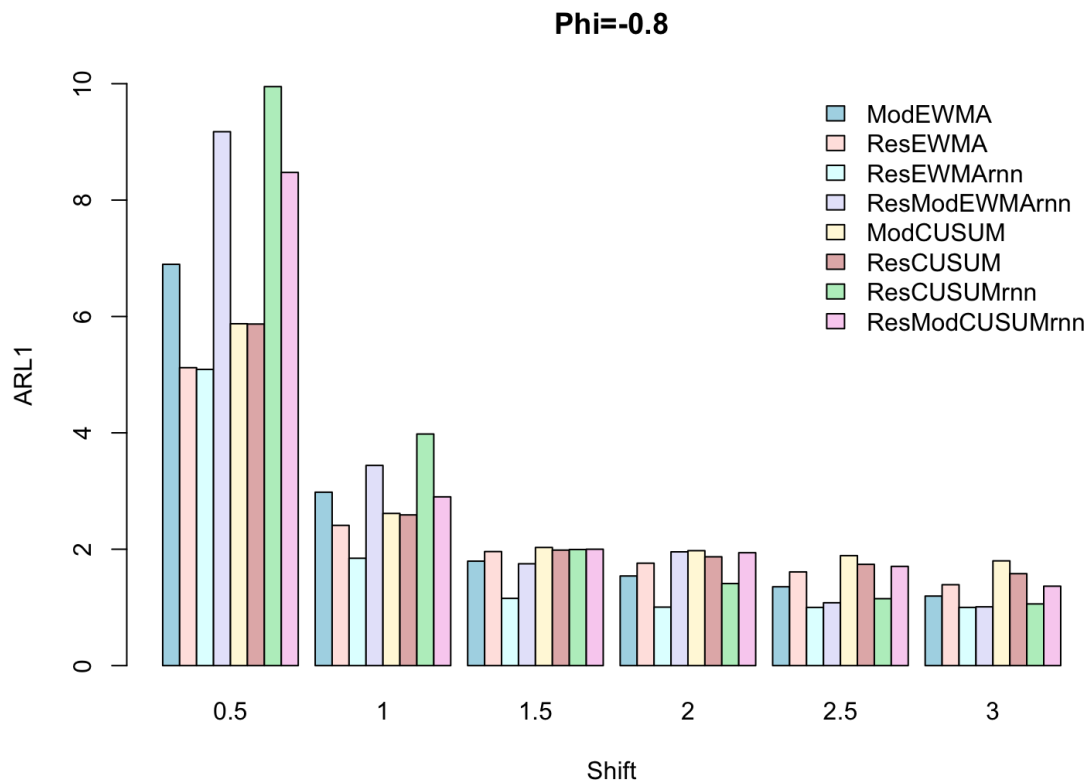
Όπως πριν, παρατηρούμε ότι τα τροποποιημένα διαγράμματα ModEWMA, ModCUSUM έχουν καλύτερη απόδοση από τα απλά διαγράμματα υπολοίπων ResEWMA, ResCUSUM για μικρή μετατόπιση. Όσο αυτή αυξάνεται, τα διαγράμματα υπολοίπων ResEWMA και ResCUSUM, υπερνικούν τα τροποποιημένα. Βέβαια, την καλύτερη απόδοση για κάθε είδους μετατόπισης έχουν ξανά τα διαγράμματα υπολοίπων του νευρωνικού δικτύου με εκείνο του EWMA να επιτυγχάνει την μικρότερη τιμή  $ARL_1$ .





Σχήμα 5.3:  $ARL_1$  για συσχέτιση  $\phi_1 = -0.2$  στα διάφορα διαγράμματα

Παρατηρούμε ότι πλέον τα απλά διαγράμματα υπολοίπων αποδίδουν καλύτερα από τα τροποποιημένα ακόμα και σε μικρή μετατόπιση. Από τα EWMA, μικρότερη τιμή  $ARL_1$  επιτυγχάνει το ResEWMArnn για όλες τις μετατοπίσεις. Το αντίστοιχο ResCUSUMrnn, φαίνεται ότι δε λειτουργεί καθόλου καλά για μικρή μετατόπιση, ενώ για μεγαλύτερες έχει την καλύτερη απόδοση. Τέλος, τα EWMA αντιλαμβάνονται πάλι γρηγορότερα τις μετατοπίσεις από τα αντίστοιχα CUSUM.



Σχήμα 5.4:  $ARL_1$  για συσχέτιση  $\phi_1 = -0.8$  στα διάφορα διαγράμματα

Σε μεγάλες αρνητικές συσχετίσεις, παρατηρούμε ότι από τα διαγράμματα υπολοίπων του νευρωνικού δικτύου το μόνο που έχει καλή απόδοση, και μάλιστα την καλύτερα από όλα τα άλλα διαγράμματα και για όλες τις μετατοπίσεις, είναι το ResEWMArnn. Τα άλλα διαγράμματα του νευρωνικού για μετατοπίσεις 0.5, 1 έχουν την χειρότερη απόδοση. Επίσης, παρατηρούμε ότι το τροποποιημένο ModEWMA ανταποκρίνεται καλύτερα από το απλό διάγραμμα υπολοίπων ResEWMA για μετατοπίσεις μεγαλύτερες από 1. Από τα CUSUM, για μεγαλύτερες μετατοπίσεις επικρατεί το ResCUSUMrnn.

## 5.2 Σύγκριση για διαδικασία $ARIMA(1, 0, 1)$

Θεωρούμε μία διαδικασία-στόχο  $\{Y_t\}$  με μέση τιμή  $\mu_0$  και διασπορά  $\sigma_0^2$ , η οποία τώρα ακολουθεί το μοντέλο  $ARIMA(1, 0, 1)$  ή πιο απλά  $ARMA(1,1)$ . Η εξίσωση που περιγράφει το μοντέλο από το οποίο προσομοιώσαμε τιμές είναι το εξής:

$$Y_t = \phi_1 Y_{t-1} + \theta_1 \epsilon_{t-1} + \epsilon_t$$

με  $\epsilon_t \sim WN(0, \sigma^2)$ .

Οι περιπτώσεις που εξετάστηκαν είναι 2. Στην πρώτη έχουμε  $\phi_1 = 0.5$  και  $\theta_1 = 0.5$ , ενώ στη δεύτερη  $\phi_1 = -0.5$  και  $\theta_1 = -0.5$ .

Εκπαιδεύτηκαν 2 νευρωνικά δίκτυα, ένα για την αρνητική και ένα για τη θετική συσχέτιση.

Αφού εκπαιδευτεί το δίκτυο, προσομοιώνουμε 1000 χρονοσειρές από το μοντέλο  $ARIMA(1, 0, 1)$  με την συσχέτιση που μας ενδιαφέρει κάθε φορά και κάνοντας τις προβλέψεις μέσω του RNN για κάθε μία από αυτές, λαμβάνουμε τα υπόλοιπα.

Η μοναδική αλλαγή στην αρχιτεκτονική του δικτύου σε σύγκριση με αυτό της  $AR(1)$  διαδικασίας είναι ότι αυξήθηκε ο αριθμός των κόμβων των κρυφών στρωμάτων από 60 σε 70, διότι αυξήθηκε ελάχιστα η πολυπλοκότητα των δεδομένων. Επίσης το Drop rate τέθηκε ίσο με 0.15.

Τα υπόλοιπα του νευρωνικού ήταν ανεξάρτητα και ακολουθούσαν την κανονική κατανομή.

Πίνακας 5.4: Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη  $ARL_0$  σε διαδικασία  $ARIMA(1, 0, 1)$

ModEWMA							
$ARL_0 \approx 370$		Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	$\theta_1$	0.5	1	1.5	2	2.5	3
-0.5	-0.5	$\lambda=0.35$	$\lambda=0.35$	$\lambda=0.45$	$\lambda=0.45$	$\lambda=0.55$	$\lambda=0.60$
		L=1.262	L=1.262	L=1.420	L=1.420	L=1.562	L=1.639
0.5	0.5	$\lambda=0.05$	$\lambda=0.05$	$\lambda=0.2$	$\lambda=0.3$	$\lambda=0.4$	$\lambda=0.65$
		L=2.628	L=2.628	L=2.783	L=2.708	L=2.650	L=2.447

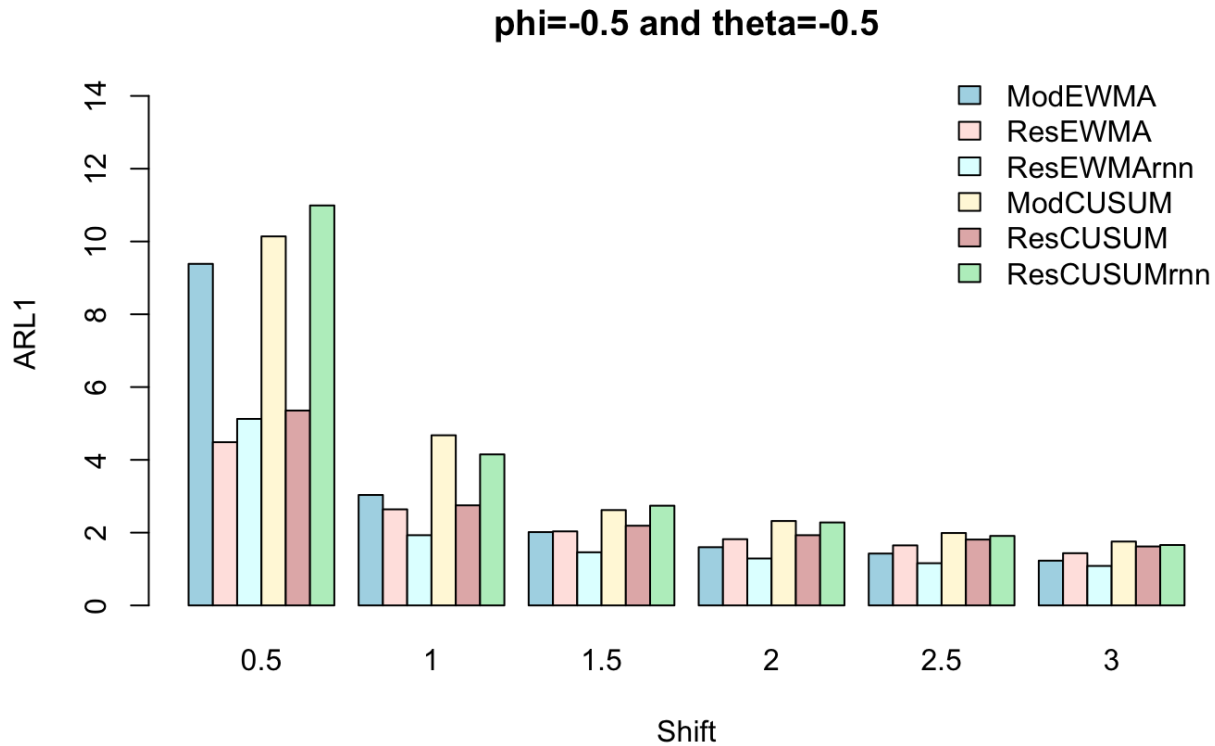
Πίνακας 5.5: Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη  $ARL_0$  σε διαδικασία  $ARIMA(1, 0, 1)$

ModCUSUM							
$ARL_0 \approx 370$		Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	$\theta_1$	0.5	1	1.5	2	2.5	3
-0.5	-0.5	k=0.25	k=0.50	k=0.75	k=1.00	k=1.25	$\lambda=1.50$
		h=2.942	h=2.253	h=1.621	h=1.224	h=1.105	h=1.086
0.5	0.5	k=0.25	k=0.50	k=0.75	k=1.00	k=1.25	k=1.5
		h=6.081	h=3.930	h=2.869	h=2.152	h=1.679	h=1.383

Πίνακας 5.6: Σύγκριση διαγραμμάτων με βάση  $ARL_1$  για μία  $ARIMA(1, 0, 1)$  διαδικασία, όταν  $ARL_0 \approx 370$

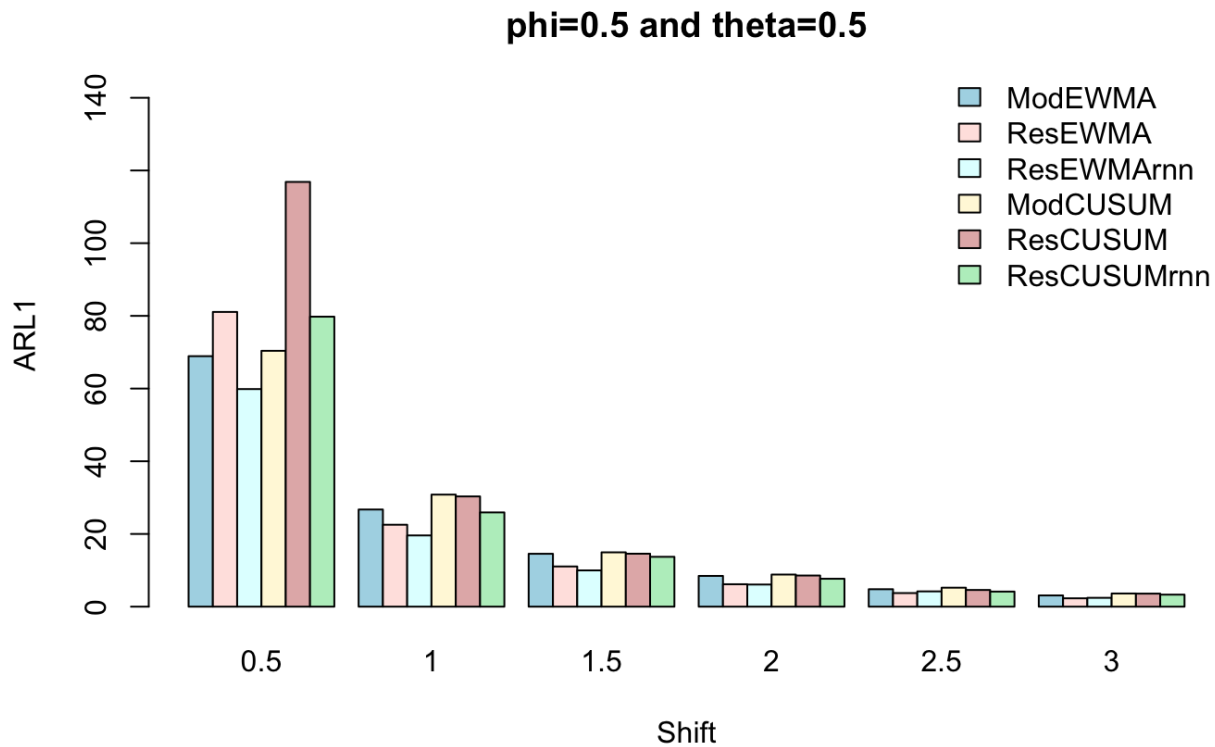
ARIMA(1, 0, 1)			Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	$\theta_1$	Μέθοδος	0.5	1	1.5	2	2.5	3
-0.5	-0.5	ModEWMA	9.385	3.035	2.015	1.600	1.425	1.230
		ModCUSUM	10.140	4.675	2.620	2.320	1.990	1.755
		ResEWMA	<b>4.485</b>	2.640	2.035	1.820	1.650	1.435
		ResCUSUM	5.355	2.750	2.190	1.930	1.810	1.620
		ResEWMArnn	5.125	<b>1.930</b>	<b>1.460</b>	<b>1.290</b>	<b>1.160</b>	<b>1.085</b>
		ResCUSUMrnn	10.990	4.150	2.740	2.280	1.910	1.660
0.5	0.5	ModEWMA	68.910	26.725	14.545	8.455	4.795	3.085
		ModCUSUM	70.385	30.825	14.940	8.830	5.215	3.620
		ResEWMA	81.080	22.535	11.055	6.165	<b>3.735</b>	<b>2.295</b>
		ResCUSUM	116.850	30.325	14.565	8.565	4.590	3.585
		ResEWMArnn	<b>59.860</b>	<b>19.595</b>	<b>9.975</b>	<b>6.085</b>	4.200	2.430
		ResCUSUMrnn	79.775	25.925	13.710	7.675	4.145	3.305

## 5.2.1 Σχολιασμός αποτελεσμάτων



Σχήμα 5.5:  $ARL_1$  για συσχέτιση  $\phi_1 = -0.5$  και  $\theta = -0.5$  στα διάφορα διαγράμματα

Η πρώτη παρατήρηση στην περίπτωση αυτή είναι ότι τα τροποποιημένα διαγράμματα ModEWMA και ModCUSUM, καθώς και το ResCUSUMrnn, δν έχουν καθόλου καλή απόδοση για μικρή μετατόπιση. Αντίθετα τα απλά διαγράμματα υπολοίπων ResEWMA και ResCUSUM, φαίνεται να αποδίδουν αρκετά καλά. Όσο αυξάνεται η μετατόπιση, το ResEWMArnn, επιτυγχάνει τη μικρότερη τιμή  $ARL_1$ . Από τα διαγράμματα CUSUM, καλύτερα ανταποκρίνεται το απλό ResCUSUM για όλες τις μετατοπίσεις.



Σχήμα 5.6:  $ARL_1$  για συσχέτιση  $\phi_1 = 0.5$  και  $\theta = 0.5$  στα διάφορα διαγράμματα

Παρατηρούμε ότι για μετατόπιση 0.5, το τροποποιημένο EWMA αποδίδει καλύτερα από το ResEWMA, αλλά καθώς το μέγεθος της μετατόπισης αυξάνεται οι ρόλοι αντιστρέφονται. Το ίδιο συμβαίνει και με τα αντίστοιχα CUSUM. Όσον αφορά τα διαγράμματα υπολοίπων από το νευρωνικό δίκτυο, το ResEWMArnn φαίνεται να υπερνικά όλα τα διαγράμματα με εξαίρεση τις μετατοπίσεις 2.5, 3 όπου καλύτερο αποδεικνύεται το ResEWMA. Το ResCUSUMrnn, αποδίδει αρκετά καλά για μετατοπίσεις μεγαλύτερες του 1.

### 5.3 Σύγκριση για διαδικασία $SARIMA(1, 0, 0)(1, 0, 0)_4$

Θεωρούμε μία διαδικασία-στόχο  $\{Y_t\}$  με μέση τιμή  $\mu_0$  και διασπορά  $\sigma_0^2$ , η οποία τώρα ακολουθεί το μοντέλο  $SARIMA(1, 0, 0)(1, 0, 0)_4$ . Δηλαδή υπάρχει επιπλέον η συνιστώσα της εποχικότητας και συγκεκριμένα με συχνότητα 4. Αυτό σημαίνει ότι λαμβάνουμε τις τιμές ανά τρίμηνο.

Συγκεκριμένα εξετάζουμε τα εξής μοντέλα:

1.  $(1 - 0.5B)(1 - 0.4B^4)Y_t = \epsilon_t$
2.  $(1 + 0.5B)(1 - 0.4B^4)Y_t = \epsilon_t$
3.  $(1 - 0.5B)(1 + 0.4B^4)Y_t = \epsilon_t$
4.  $(1 + 0.5B)(1 + 0.4B^4)Y_t = \epsilon_t$

Όλες οι παραπάνω χρονοσειρές είναι στάσιμες με  $\epsilon_t \sim WN(0, \sigma^2)$ .

Η μοναδική αλλαγή στην αρχιτεκτονική του δικτύου σε σύγκριση με αυτό της  $ARIMA(1,0,1)$  διαδικασίας είναι ότι αυξήθηκε ο αριθμός των κόμβων των κρυφών στρωμάτων από 70 σε 75, διότι αυξήθηκε ελάχιστα η πολυπλοκότητα των δεδομένων. Το Drop rate επανήλθε στο 0.20.

Εκπαιδεύτηκαν 4 νευρωνικά δίκτυα, ένα για κάθε συσχέτιση.

Αφού εκπαιδευτεί το δίκτυο, προσομοιώνουμε 1000 χρονοσειρές από το μοντέλο  $SARIMA(1, 0, 0)(1, 0, 0)_4$  με την συσχέτιση που μας ενδιαφέρει κάθε φορά και κάνοντας τις προβλέψεις μέσω του RNN για κάθε μία από αυτές, λαμβάνουμε τα υπόλοιπα.

Τα υπόλοιπα του νευρωνικού ήταν ανεξάρτητα και ακολουθούσαν την κανονική κατανομή.

Πίνακας 5.7: Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη  $ARL_0$  σε διαδικασία  $SARIMA(1, 0, 0)(1, 0, 0)_4$

ModEWMA							
$ARL_0 \approx 370$		Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	$\Phi_1$	0.5	1	1.5	2	2.5	3
-0.5	-0.4	$\lambda=0.10$ L=1.150	$\lambda=0.2$ L=1.358	$\lambda=0.35$ L=1.566	$\lambda=0.3$ L=1.514	$\lambda=0.3$ L=1.514	$\lambda=0.55$ L=1.778
-0.5	0.4	$\lambda=0.1$ L=1.509	$\lambda=0.3$ L=1.600	$\lambda=0.3$ L=1.600	$\lambda=0.6$ L=1.791	$\lambda=0.6$ L=1.791	$\lambda=0.6$ L=1.791
0.5	-0.4	$\lambda=0.05$ L=1.993	$\lambda=0.05$ L=1.993	$\lambda=0.15$ L=2.284	$\lambda=0.15$ L=2.284	$\lambda=0.3$ L=2.382	$\lambda=0.75$ L=2.275
0.5	0.4	$\lambda=0.05$ L=2.726	$\lambda=0.05$ L=2.726	$\lambda=0.15$ L=2.81	$\lambda=0.45$ L=2.574	$\lambda=0.95$ L=2.152	$\lambda=0.95$ L=2.152

Πίνακας 5.8: Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη  $ARL_0$  σε διαδικασία  $SARIMA(1, 0, 0)(1, 0, 0)_4$

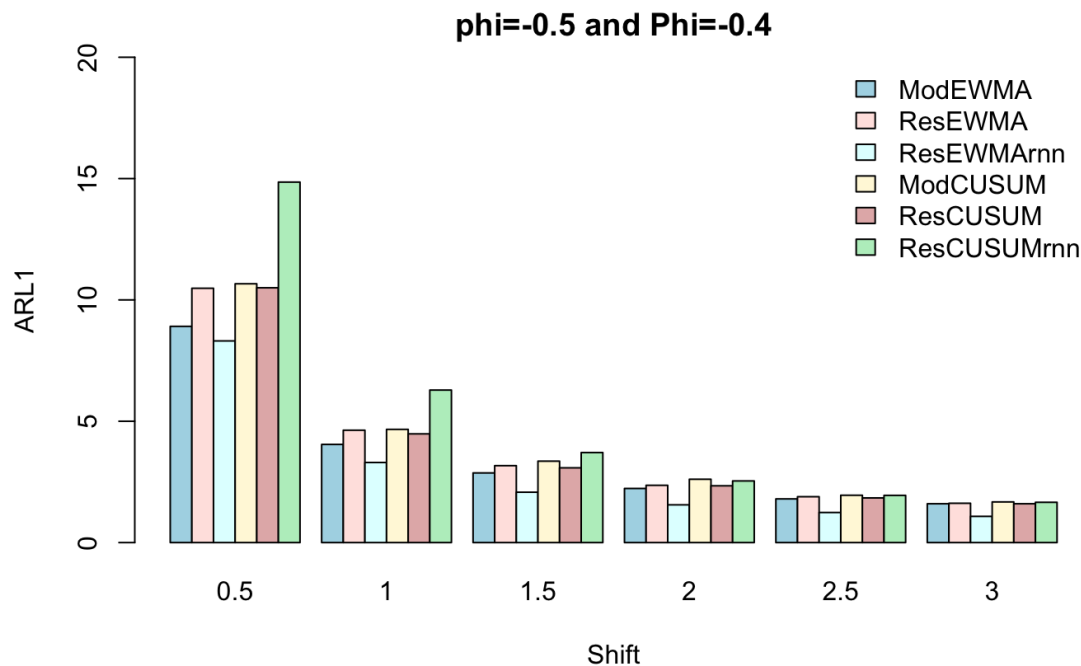
ModCUSUM							
$ARL_0 \approx 370$		Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	$\Phi_1$	0.5	1	1.5	2	2.5	3
-0.5	-0.4	k=0.25 h=3.444	k=0.50 h=2.718	k=0.75 h=2.310	k=1.00 h=2.000	k=1.25 h=1.776	k=1.50 h=1.524
-0.5	0.4	k=0.25 h=4.872	k=0.50 h=2.754	k=0.75 h=2.356	k=1.00 h=2.108	k=1.25 h=1.918	k=1.50 h=1.690
0.5	-0.4	k=0.25 h=8.251	k=0.50 h=6.963	k=0.75 h=5.233	k=1.00 h=4.114	k=1.25 h=3.116	k=1.50 h=2.324
0.5	0.4	k=0.25 h=10.449	k=0.50 h=8.961	k=0.75 h=6.938	k=1.00 h=5.209	k=1.25 h=4.317	k=1.50 h=2.936



Πίνακας 5.9: Σύγκριση διαγραμμάτων με βάση  $ARL_1$  για μία  $SARIMA(1, 0, 0)(1, 0, 0)_4$  διαδικασία, όταν  $ARL_0 \approx 370$ 

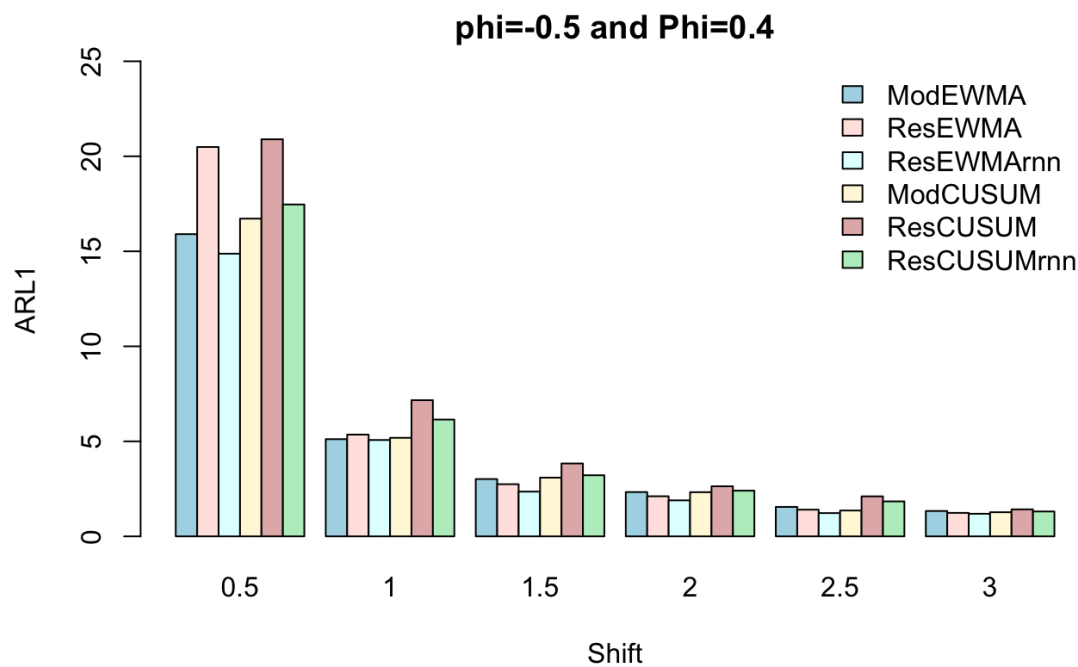
SARIMA(1, 0, 0)(1, 0, 0) <sub>4</sub>			Μετατόπιση (κλίμακα $\sigma$ )					
$\phi_1$	$\Phi_1$	Μέθοδος	0.5	1	1.5	2	2.5	3
-0.5	-0.4	ModEWMA	8.910	4.045	2.870	2.230	1.800	1.600
		ModCUSUM	10.665	4.665	3.355	2.610	1.950	1.675
		ResEWMA	10.480	4.630	3.170	2.360	1.890	1.620
		ResCUSUM	10.500	4.480	3.080	2.340	1.840	1.600
		ResEWMArnn	<b>8.310</b>	<b>3.300</b>	<b>2.075</b>	<b>1.555</b>	<b>1.240</b>	<b>1.080</b>
		ResCUSUMrnn	14.855	6.285	3.710	2.540	1.945	1.660
-0.5	0.4	ModEWMA	15.905	5.115	3.015	2.330	1.550	1.335
		ModCUSUM	16.720	5.185	3.090	2.325	1.360	1.270
		ResEWMA	20.490	5.355	2.745	2.110	1.405	1.235
		ResCUSUM	20.895	7.165	3.835	2.635	2.105	1.420
		ResEWMArnn	<b>14.875</b>	<b>5.070</b>	<b>2.355</b>	<b>1.895</b>	<b>1.225</b>	<b>1.190</b>
		ResCUSUMrnn	17.465	6.145	3.215	2.405	1.840	1.310
0.5	-0.4	ModEWMA	30.735	13.915	7.325	5.205	3.835	3.015
		ModCUSUM	35.715	13.730	7.910	4.880	3.375	2.370
		ResEWMA	35.305	10.690	5.580	3.555	2.405	1.610
		ResCUSUM	38.350	12.825	5.670	3.450	<b>2.245</b>	<b>1.410</b>
		ResEWMArnn	<b>25.405</b>	<b>8.440</b>	<b>4.870</b>	<b>3.405</b>	2.465	1.935
		ResCUSUMrnn	38.205	12.510	6.985	4.680	3.175	2.335
0.5	0.4	ModEWMA	100.290	31.830	18.115	8.805	3.780	2.090
		ModCUSUM	119.395	37.825	17.720	8.745	3.395	2.080
		ResEWMA	117.945	36.215	12.830	<b>6.730</b>	3.685	2.005
		ResCUSUM	120.335	42.080	17.855	8.630	<b>3.350</b>	<b>1.420</b>
		ResEWMArnn	<b>85.310</b>	<b>27.965</b>	<b>12.650</b>	7.645	4.905	3.035
		ResCUSUMrnn	110.08	33.840	13.675	7.780	5.140	3.775

## 5.3.1 Σχολιασμός αποτελεσμάτων



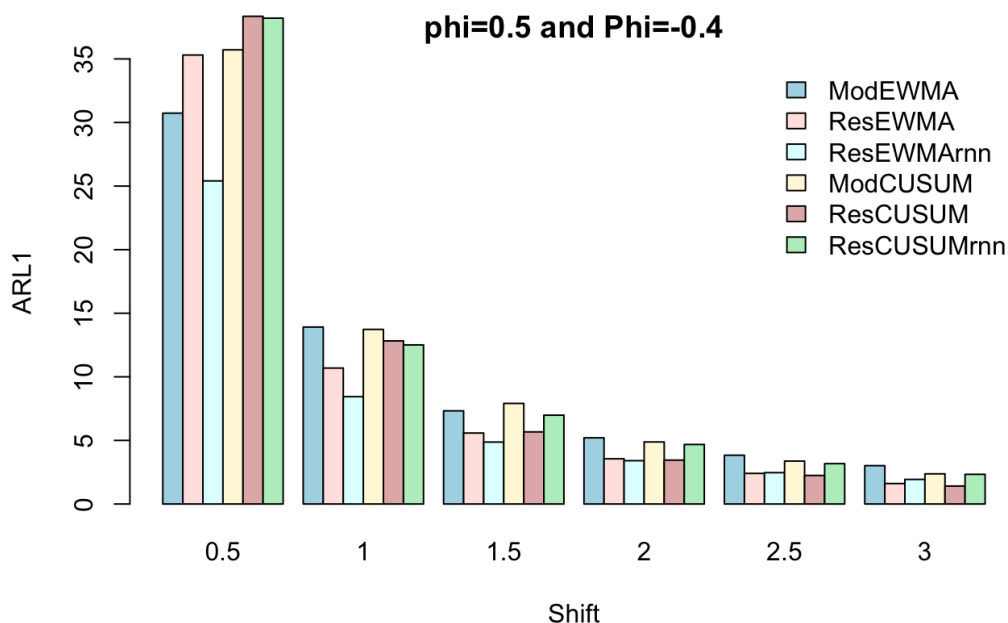
Σχήμα 5.7:  $ARL_1$  για συσχέτιση  $\phi_1 = -0.5$  και  $\Phi_1 = -0.4$  στα διάφορα διαγράμματα

Παρατηρούμε ότι για  $\phi_1 = -0.5$  και  $\Phi_1 = -0.4$  από τα EWMA καλύτερη απόδοση έχει το ResEWMArnn, μετά το ModEWMA και τελευταίο το ResEWMA. Μάλιστα το ResEWMArnn υπερσχύει έναντι όλων των διαγραμμάτων για κάθε είδους μετατόπισης. Από τα CUSUM, την χειρότερη απόδοση φαίνεται να έχει το ResCUSUMrnn, ειδικά για μικρή μετατόπιση. Ανάμεσα στα ModCUSUM και ResCUSUM, το δεύτερο φαίνεται να αποδίδει λίγο καλύτερα.



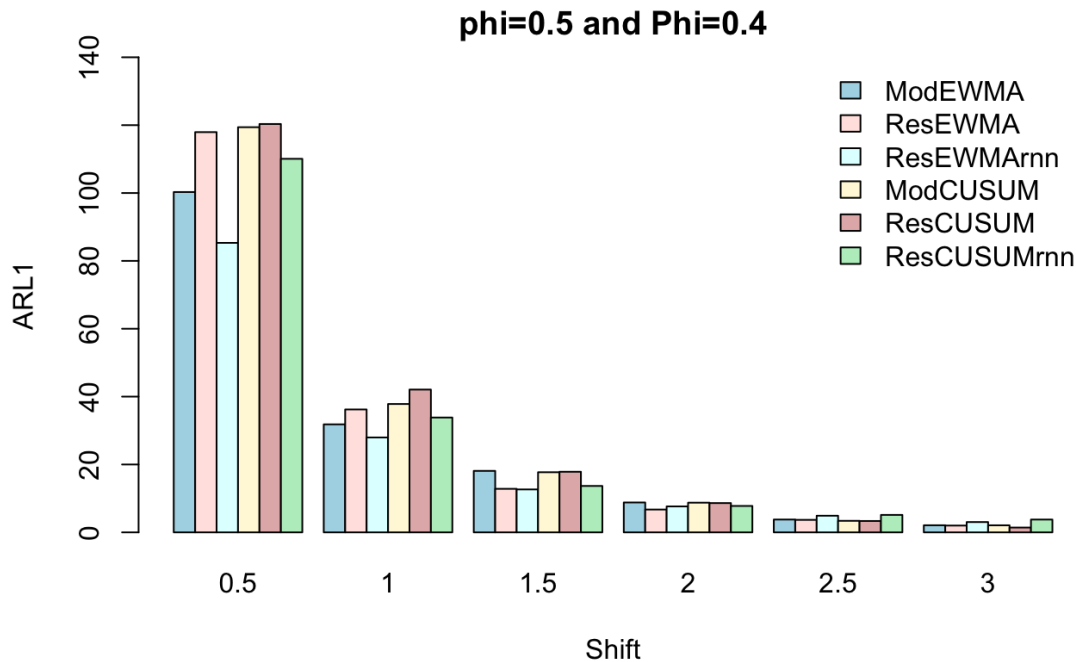
Σχήμα 5.8:  $ARL_1$  για συσχέτιση  $\phi_1 = -0.5$  και  $\Phi_1 = 0.4$  στα διάφορα διαγράμματα

Όσον αφορά τα EWMA, για τις δύο πρώτες μετατοπίσεις, το ModEWMA αποδίδει καλύτερα από το ResEWMA, αλλά από εκεί και έπειτα συμβαίνει το αντίστροφο. Το ResEWMArnn έχει πάλι την καλύτερη απόδοση από όλα τα διαγράμματα. Από τα CUSUM, φαίνεται πως καλύτερη απόδοση απ' όλα έχει το ModCUSUM και χειρότερη το ResCUSUM.



Σχήμα 5.9:  $ARL_1$  για συσχέτιση  $\phi_1 = 0.5$  και  $\Phi_1 = -0.4$  στα διάφορα διαγράμματα

Παρατηρούμε ξανά ότι το ResEWMArnn έχει την καλύτερη απόδοση από τα υπόλοιπα διαγράμματα, με εξαίρεση τις μετατοπίσεις 2.5 και 3, όπου υπερσχύει το ResCUSUM. Τα τροποποιημένα διαγράμματα δε φαίνεται να έχουν τόσο καλή απόδοση, με εξαίρεση τη μετατόπιση 0.5.



Σχήμα 5.10:  $ARL_1$  για συσχέτιση  $\phi_1 = 0.5$  και  $\Phi_1 = 0.4$  στα διάφορα διαγράμματα

Για θετικές συσχετίσεις  $\phi_1 = 0.5$  και  $\Phi_1 = 0.4$ , το ResEWMArnn φαίνεται να έχει την καλύτερη απόδοση μέχρι την μετατόπιση 1.5. Για μετατόπιση 2 υπερσχύει το ResEWMA, ενώ για τις υπόλοιπες το ResCUSUM. Από τα διαγράμματα CUSUM, το ResCUSUMrnn φαίνεται να ανταποκρίνεται καλύτερα σε μετατοπίσεις μέχρι  $2\sigma_0$ . Το ModEWMA έχει καλύτερη απόδοση από το ResEWMA μόνο για τις δύο πρώτες μετατοπίσεις.

### 5.4 Σύγκριση για διαδικασία ETS (A,N,N)

Θεωρούμε μία διαδικασία-στόχο  $\{Y_t\}$  με μέση τιμή  $\mu_0$  και διασπορά  $\sigma_0^2$ , η οποία τώρα ακολουθεί το μοντέλο ETS(A,N,N), δηλαδή το Simple Exponential Smoothing όπως περιγράφηκε στο κεφάλαιο 2.

Ο όρος «A» αναφέρεται στην λέξη *Additive* και περιγράφει το είδος των σφαλμάτων του μοντέλου. Το πρώτο «N» μας πληροφορεί ότι δεν υπάρχει κάποιο συγκεκριμένο είδος τάσης στην χρονοσειρά, ενώ το δεύτερο «N» (None) αναφέρεται στην απουσία εποχικότητας. Δυστυχώς, τα μοντέλα ETS δεν περιγράφουν στάσιμες χρονοσειρές, οπότε πριν εφαρμόσουμε σε αυτά τα διαγράμματα ελέγχου την μετατρέπουμε σε στάσιμη λαμβάνοντας μία διαφορά.

Η εξισώσεις που περιγράφουν το παραπάνω στατιστικό μοντέλο είναι οι εξής:

$$\begin{aligned}y_t &= l_{t-1} + \epsilon_t \\l_t &= l_{t-1} + \alpha \epsilon_t\end{aligned}$$

, όπου  $l_t$  η εξομαλυμένη τιμή της  $Y_t$  τη χρονική στιγμή  $t$ . Επειδή τα σφάλματα είναι αθροιστικά (additive) θα υπολογίζονται ως:

$$\epsilon_t = y_t - l_{t-1} = y_t - \hat{y}_{t|t-1} \sim WN(0, \sigma^2)$$

Οι προσομοιωμένες τιμές ελήφθησαν για  $\alpha = 0.5$ . Το ETS(A,N,N) είναι ισοδύναμο με μία διαδικασία ARIMA(0,1,1) με  $\theta_1 = \alpha - 1 = -0.5$ .

Η διαφορά από την αρχιτεκτονική του δικτύου της SARIMA(1, 0, 0)(1, 0, 0)<sub>4</sub> διαδικασίας είναι ότι μειώθηκε ο αριθμός των κόμβων των κρυφών στρώματων από 75 σε 60. Το Drop rate παρέμεινε 0.2. Στο δίκτυο εισάγουμε τα δεδομένα αφού πρώτα τα έχουμε μετασχηματίσει σε στάσιμα. Ο λόγος που γίνεται αυτό, είναι για να μειώσουμε την πολυπλοκότητα του νευρωνικού δικτύου, καθώς διαφορετικά θα χρειαζόνταν περισσότερα στρώματα και εποχές για την εκπαίδευση του.

Αφού εκπαιδευτεί το δίκτυο, προσομοιώνουμε 1000 χρονοσειρές από το μοντέλο ETS(A,N,N) με  $\alpha = 0.5$  και κάνοντας τις προβλέψεις μέσω του RNN για κάθε μία από αυτές, λαμβάνουμε τα υπόλοιπα.

Πίνακας 5.10: Βέλτιστοι συνδυασμοί των παραμέτρων του ModEWMA για επίτευξη  $ARL_0 \approx 370$  σε μία διαδικασία ETS(A,N,N)

ModEWMA					
Μετατόπιση (κλίμακα $\sigma$ )					
0.5	1	1.5	2	2.5	3
$\lambda=0.05$ L=1.814	$\lambda=0.15$ L=1.983	$\lambda=0.3$ L=2.099	$\lambda=0.3$ L=2.099	$\lambda=0.75$ L=2.125	$\lambda=0.75$ L=2.125

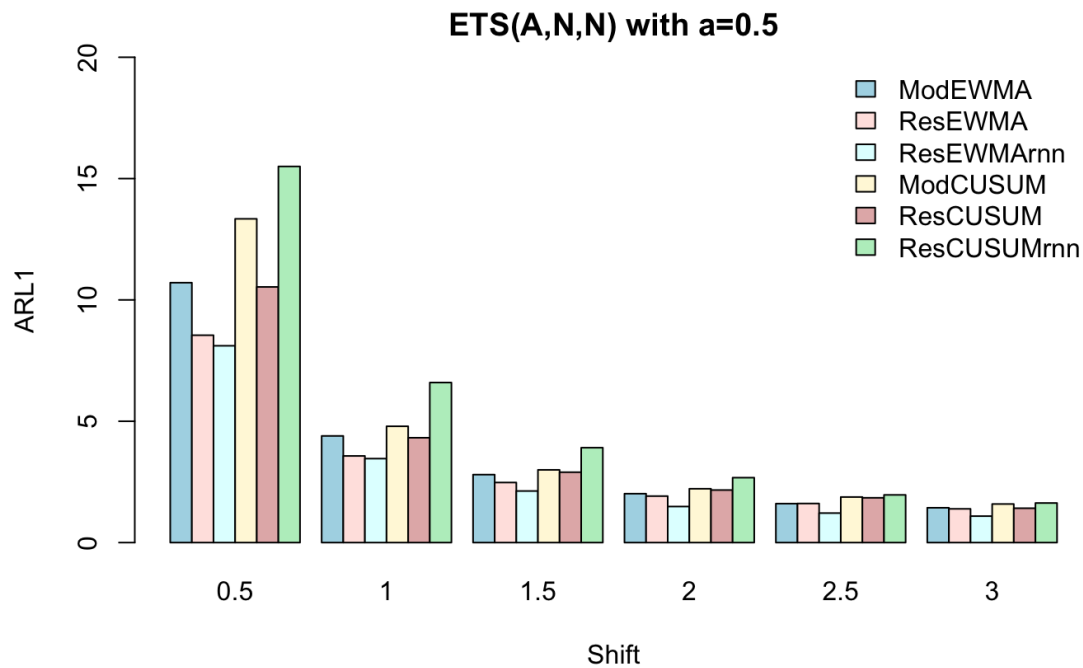
Πίνακας 5.11: Βέλτιστοι συνδυασμοί των παραμέτρων του ModCUSUM για επίτευξη  $ARL_0 \approx 370$  σε μία διαδικασία ETS(A,N,N)

ModCUSUM					
Μετατόπιση (κλίμακα $\sigma$ )					
0.5	1	1.5	2	2.5	3
k=0.25 h=4.531	k=0.50 h=4.319	k=0.75 h=4.019	k=1.00 h=3.754	k=1.25 h=3.501	k=1.50 h=3.272

Πίνακας 5.12: Σύγκριση διαγραμμάτων με βάση  $ARL_1$  για μία ETS(A,N,N) διαδικασία, όταν  $ARL_0 \approx 370$

ETS(A,N,N) Μέθοδος	Μετατόπιση (κλίμακα $\sigma$ )					
	0.5	1	1.5	2	2.5	3
ModEWMA	10.710	4.395	2.800	2.015	1.605	1.435
ModCUSUM	13.341	4.792	2.996	2.218	1.877	1.589
ResEWMA	8.545	3.570	2.475	1.915	1.610	1.390
ResCUSUM	10.535	4.320	2.900	2.165	1.845	1.415
ResEWMArnn	<b>8.110</b>	<b>3.460</b>	<b>2.125</b>	<b>1.485</b>	<b>1.215</b>	<b>1.090</b>
ResCUSUMrnn	15.500	6.595	3.910	2.675	1.965	1.630

## 5.4.1 Σχολιασμός αποτελεσμάτων



Σχήμα 5.11:  $ARL_1$  για το μοντέλο ETS(A,N,N) με  $\alpha = 0.5$  στα διάφορα διαγράμματα

Παρατηρούμε ότι την καλύτερη απόδοση από όλα τα διαγράμματα έχει το ResEWMArnn και αμέσως μετά το ResEWMA. Τα τροποποιημένα διαγράμματα φαίνεται να μην αποδίδουν τόσο καλά όσο τα απλά διαγράμματα υπολοίπων. Το ResCUSUMrnn ανταποκρίνεται χειρότερα για όλες τις μετατοπίσεις.

## 5.5 Σύνοψη αποτελεσμάτων

### Γενικές παρατηρήσεις

- Το ResEWMArnn στις περισσότερες περιπτώσεις φαίνεται να έχει την καλύτερη απόδοση.
- Για αρνητικές συσχετίσεις όλα τα διαγράμματα ανεξαιρέτως αποδίδουν πολύ καλύτερα απ' ότι στις θετικές.
- Τα EWMA διαγράμματα έχουν στο σύνολο των περιπτώσεων που εξετάζουμε καλύτερη απόδοση από τα αντίστοιχα CUSUM.
- Το ResCUSUMrnn για αρνητική συσχέτιση και μικρή μετατόπιση έχει μεγαλύτερη διαφορά στην απόδοση (χειρότερη) από τα υπόλοιπα διαγράμματα.
- Στο μοντέλο SARIMA(1, 0, 0)(1, 0, 0)<sub>4</sub> η αρνητική συσχέτιση στον όρο  $\phi_1$  επιφέρει καλύτερα αποτελέσματα απ' ότι η αρνητική συσχέτιση στον όρο  $\Phi_1$  του εποχιακού μέρους του μοντέλου.

### AR(1)

- Για θετικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των EWMA είναι η εξής : ResEWMArnn > ModEWMA > ResEWMA. Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.
- Για θετικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των CUSUM είναι η εξής : ResCUSUMrnn > ModCUSUM > ResCUSUM. Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.
- Για αρνητικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των EWMA είναι η εξής : ResEWMArnn > ResEWMA > ModEWMA . Για μεγαλύτερες μετατοπίσεις και πολύ αρνητικές συσχετίσεις προτιμώνται τα τροποποιημένα διαγράμματα, ενώ για λιγότερο αρνητικές τα διαγράμματα υπολοίπων.



- Για αρνητικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των CUSUM είναι η εξής :  $\text{ResCUSUM} > \text{ModCUSUM} > \text{ResCUSUMrnn}$ . Για μεγαλύτερες μετατοπίσεις προτιμάται το  $\text{ResCUSUMrnn}$ .

### ARIMA(1,0,1)

- Για θετικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των EWMA είναι η εξής :  $\text{ResEWMArnn} > \text{ModEWMA} > \text{ResEWMA}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.

- Για θετικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των CUSUM είναι η εξής :  $\text{ModCUSUM} > \text{ResCUSUMrnn} > \text{ResCUSUM}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.

- Για αρνητικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των EWMA είναι η εξής :  $\text{ResEWMA} > \text{ResEWMArnn} > \text{ModEWMA}$ . Για μεγαλύτερες μετατοπίσεις προτιμάται το  $\text{ResEWMArnn}$ .

- Για αρνητικές συσχετίσεις και μικρή μετατόπιση η προτίμηση των CUSUM είναι η εξής :  $\text{ModCUSUM} > \text{ResCUSUMrnn} > \text{ResCUSUM}$ . Για μεγαλύτερες μετατοπίσεις προτιμάται το  $\text{ResCUSUMrnn}$ .

### SARIMA(1, 0, 0)(1, 0, 0)<sub>4</sub>

- Αν  $\phi_1, \Phi_1 > 0$  η προτίμηση των EWMA για μικρή μετατόπιση είναι η εξής:  $\text{ResEWMArnn} > \text{ModEWMA} > \text{ResEWMA}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.

- Αν  $\phi_1, \Phi_1 > 0$  η προτίμηση των CUSUM για μικρή μετατόπιση είναι η εξής :  $\text{ResCUSUMrnn} > \text{ModCUSUM} > \text{ResCUSUM}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.

- Αν  $\phi_1 > 0$  και  $\Phi_1 < 0$  η προτίμηση των EWMA για μικρή μετατόπιση είναι η εξής :  $\text{ResEWMArnn} > \text{ModEWMA} > \text{ResEWMA}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.

- Αν  $\phi_1 > 0$  και  $\Phi_1 < 0$  η προτίμηση των CUSUM για μικρή μετατόπιση είναι η εξής :  $\text{ModCUSUM} > \text{ResCUSUMrnn} > \text{ResCUSUM}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.
- Αν  $\phi_1 < 0$  και  $\Phi_1 > 0$  η προτίμηση των EWMA για μικρή μετατόπιση είναι η εξής :  $\text{ResEWMArnn} > \text{ModEWMA} > \text{ResEWMA}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα διαγράμματα υπολοίπων.
- Αν  $\phi_1 < 0$  και  $\Phi_1 > 0$  η προτίμηση των CUSUM για μικρή μετατόπιση είναι η εξής :  $\text{ModCUSUM} > \text{ResCUSUMrnn} > \text{ResCUSUM}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα ModCUSUM και ResCUSUMrnn.
- Αν  $\phi_1, \Phi_1 < 0$  η προτίμηση των EWMA για μικρή μετατόπιση είναι η εξής :  $\text{ResEWMArnn} > \text{ModEWMA} > \text{ResEWMA}$ . Για μεγαλύτερες μετατοπίσεις προτιμώνται τα ModEWMA και ResEWMArnn.
- Αν  $\phi_1, \Phi_1 < 0$  η προτίμηση των CUSUM για μικρή μετατόπιση είναι η εξής :  $\text{ResCUSUM} > \text{ModCUSUM} > \text{ResCUSUMrnn}$ . Για μεγαλύτερες μετατοπίσεις προτιμάται το ResCUSUM.

### ETS(A,N,N)

Για το μοντέλο ETS(A,N,N) με  $\alpha = 0.5$ :

- Η προτίμηση των EWMA για κάθε μετατόπιση είναι η εξής :  $\text{ResEWMArnn} > \text{ResEWMA} > \text{ModEWMA}$ .
- Η προτίμηση των CUSUM για κάθε μετατόπιση είναι η εξής :  $\text{ResCUSUM} > \text{ModCUSUM} > \text{ResCUSUMrnn}$ .

## 5.6 Επίλογος και μελλοντική έρευνα

Στην παρούσα εργασία συνδυάστηκε η χρήση των αναδρομικών νευρωνικών δικτύων με τα διαγράμματα του στατιστικού ελέγχου διεργασιών με σκοπό την βελτίωση των κλασικών μεθόδων πάνω σε αυτοσυσχετισμένα δεδομένα. Φάνηκε πως ο συνδυασμός RNN και EWMA επέφερε τα καλύτερα αποτελέσματα στο σύνολο των περιπτώσεων που μελετήσαμε αποδεικνύοντας έτσι πως η χρήση νευρωνικών δικτύων στον Στατιστικό Έλεγχο Ποιότητας μπορεί να επιφέρει βελτιώσεις. Ενδιαφέρον θα είχε η χρήση GRUs αντί για LSTMs καθώς και η χρήση κάποιου Bi-Directional νευρωνικού δικτύου. Ακόμα, θα μπορούσε να γίνει χρήση κάποιου άλλου είδους νευρωνικών δικτύων όπως Transformer ή TCN (Temporal Convolutional Network). Η πιο περίπλοκη αρχιτεκτονική θα είναι σε θέση να αναλύει και να γενικεύει καλύτερα κατηγορίες μοντέλων με περισσότερες παραμέτρους ξεπερνώντας τελικά τα κλασικά διαγράμματα ελέγχου του SPC.

## Βιβλιογραφία

- [1] Chen S. and Yu J. Deep recurrent neural network-based residual control chart for autocorrelated processes. *Qual Reliab Engng Int.* 2019;35:2687–2708.
- [2] Chiu C.-C., Chen M.-K., & Lee K.-M. Shifts recognition in correlated process data using a neural network. *International Journal of Systems Science*, (2001) 32(2), 137–143.
- [3] Cho K. et al. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *arXiv preprint arXiv: 1406.1078*, 2014a.
- [4] Elman J.-L.(1990). Finding Structure in Time. *Cognitive Science* 14 (2): 179-211
- [5] Fathi O., Time series forecasting using a hybrid ARIMA and LSTM model.
- [6] Guh R.-S. Real-time recognition of control chart patterns in autocorrelated processes using a learning vector quantization network-based approach. *International Journal of Production Research*,(2008), 46:14, 3959-3991
- [7] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*. 9. 1735-80.
- [8] Hewamalage H., Bergmeir C., Bandara K., Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*,37 (1) ,388-427, (2021)
- [9] Holt, C. E. (1957). Forecasting seasonals and trends by exponentially weighted averages (O.N.R. Memorandum No. 52). *Carnegie Institute of Technology*, Pittsburgh USA.
- [10] Hwarng, H. B. Detecting process mean shift in the presence of autocorrelation: a neural network based monitoring scheme. *International Journal of Production Research*,(2004) 42(3), 573–595.
- [11] Kramer H. G.(1997) On Control Charts for Time Series. *Dissertation*, Universität Ulm.
- [12] Loredó E. N., Jeakrkaporn D.and Borrór C. M.. Model-based control chart for autoregressive and correlated data. *Qual. Reliab. Engng. Int.* 2002; 18: 489–496
- [13] Box G.E.P., Jenkins G., Time Series Analysis, Forecasting and Control, *Holden-Day*, San Francisco, CA, 1970.
- [14] Statistical Quality Control. Koukouvinos Christos, NTUA (2020)
- [15] Lucas, J. and Saccucci M, Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements. *Technometrics* 32, no. 1 (1990): 1-12.

- [16] Montgomery, D. C. (2013). Introduction to statistical quality control. Hoboken, NJ: Wiley.
- [17] Mulder WD, Bethard S, Moens MF. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech and Language*. 2015;30(1):61-98.
- [18] Nikiforov I.V. ,Sequential Detection of Abrupt Changes in Time Series Properties, *Nauke*, Moscow (1983)
- [19] Pacella M.and Semeraro Q, Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring. *Computers & Industrial Engineering* 52 (2007) 502–520.
- [20] Page E. S. , Continuous Inspection Schemes, *Biometrika*, Volume 41, Issue 1-2, June 1954, Pages 100–115
- [21] Purintrapiban U. and Corley H.W. , Neural networks for detecting cyclic behavior in autocorrelated process. *Computers & Industrial Engineering* 62 (2012) 1093–1108
- [22] Qiu P. , Li W.and Li J. , A New Process Control Chart for Monitoring Short- Range Serially Correlated Data *Technometrics*, (2020), 62:1, 71-83
- [23] Rius A., Ruisanchez I., Callao M.P. and Rius F.X. , Reliability of analytical systems: use of control charts, time series models and recurrent neural networks (RNN). *Chemometrics and Intelligent Laboratory Systems* 40 (1998) 1–18
- [24] Roberts S.W. (1959), Control Chart Tests Based on Geometric Moving Averages. *Technometrics*, 1, 239-250.
- [25] Rosenblatt F. (1958), The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65 6: 386-408
- [26] Sheehan S, Song YS, Deep Learning for Population Genetic Inference. *PLoS Comput Biol.* , 12(3):e1004845. (2016)
- [27] Schmid W. (1997a), On EWMA charts for time series. In: H.-J. Lenz & P.-Th. Wilrich (Eds.), *Frontiers in Statistical Quality Control*, Physica, Heidelberg, Germany, 115-137.
- [28] Schmid W. (1997b) , CUSUM control schemes for Gaussian processes. *Statistical Papers* 38, 191.
- [29] Schmid W. and Schöne A. , Some properties of the EWMA control chart in the presence of autocorrelation. *The Annals of Statistics* ,1997, Vol. 25, No. 3, 1277–1283.

- [30] Senneset K. and Gultvedt M., Something Old, Something New: A Hybrid Approach with ARIMA and LSTM to Increase Portfolio Stability. (2021).
- [31] Siami Namini S. and Tavakoli N. and Siami Namin A., A Comparison of ARIMA and LSTM in Forecasting Time Series. 1394-1401, (2018)
- [32] Wang C., Du W., Zhu Z., Yue Z., The real-time big data processing method based on LSTM or GRU for the smart job shop production process. *Journal of Algorithms & Computational Technology*, January 2020.
- [33] Wardell DG., Moskowitz H., Plante RD. Control charts in the presence of data correlation. *Management Science*. 1992; 38(8):1084-1105.
- [34] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6, 324–342.
- [35] Wu B. and Yu J., A neural network ensemble model for on-line monitoring of process mean and variance shifts in correlated processes. *Expert Systems with Applications* 37 (2010) 4058–4065
- [36] Yashchin E. (1993), Performance of CUSUM Control Schemes for Serially Correlated Observations. *Technometrics* 35(1), 37-52.