



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τηλεπικοινωνιών και Τεχνολογίας Πληροφορίας
Εργαστήριο Θεωρητικής Ηλεκτροτεχνίας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του φοιτητή του ΔΠΜΣ-Ολοκληρωμένα Συστήματα Υλικού και Λογισμικού
του Τμήματος Μηχανικών Η/Υ και Πληροφορικής
της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Νικόλαου Τουλγαρίδη

Θέμα

**«Σύστημα Μηχανικής Μάθησης
για την Αυτόματη
Αναγνώριση Χειρόγραφων Ψηφίων
βασισμένο σε υψηλής απόδοσης
RBM Hardware Accelerator»**

Επιβλέπων

Θεόδωρος Αντωνακόπουλος

Πάτρα, Φεβρουάριος 2018

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με θέμα

**«Σύστημα Μηχανικής Μάθησης
για την Αυτόματη
Αναγνώριση Χειρόγραφων Ψηφίων
βασισμένο σε υψηλής απόδοσης
RBM Hardware Accelerator»**

Του φοιτητή του Τμήματος Μηχανικών Η/Υ και
Πληροφορικής της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Τουλγαρίδη Νικόλαου
Αριθμός Μητρώου: 1003691

Παρουσιάστηκε δημόσια και εξετάστηκε στο Τμήμα Ηλεκτρολόγων
Μηχανικών και Τεχνολογίας Υπολογιστών στις 26/02/2018

Ο Επιβλέπων
Θεόδωρος Αντωνακόπουλος, Καθηγητής

Αριθμός Διπλωματικής Εργασίας:

Θέμα:

«Σύστημα Μηχανικής Μάθησης
για την Αυτόματη
Αναγνώριση Χειρόγραφων Ψηφίων
βασισμένο σε υψηλής απόδοσης
RBM Hardware Accelerator»

Φοιτητής:

Τουλγαρίδης Νικόλαος

Επιβλέπων:

Θεόδωρος Αντωνακόπουλος

Περίληψη

Η ταχεία αύξηση του μεγέθους των δεδομένων και της προσβασιμότητας τα τελευταία χρόνια έχει προκαλέσει αλλαγή στη φιλοσοφία σχεδιασμού συστημάτων με κατεύθυνση τη χρήση αλγορίθμων τεχνητής νοημοσύνης.

Τα συστήματα αυτά έχουν την ικανότητα να μαθαίνουν και να επεξεργάζονται τεράστιες ποσότητες δεδομένων ανοίγοντας νέους ορίζοντες σε σημαντικούς τομείς όπως η υπολογιστική όραση, η αναγνώριση προτύπων, η επεξεργασία φυσικής γλώσσας κ.λ.π.

Η πιο δημοφιλής κατηγορία τεχνικών που χρησιμοποιούν τεχνητά νευρωνικά δίκτυα ονομάζεται Deep Learning Networks και έχει μεγάλο εύρος εφαρμογής τόσο στην καθημερινότητα όσο και στη βιομηχανία. Ωστόσο, αυτά τα μοντέλα έχουν πολύ υψηλές απαιτήσεις σε αποθήκευση δεδομένων και υπολογιστική ισχύ, ιδιαίτερα κατά τη διαδικασία της εκπαίδευσης (training).

Λύση στο σύνθετο αυτό πρόβλημα των πόρων προσπαθεί να δοθεί με τη χρήση συστημάτων πολυεπεξεργασίας (Multiprocessors, GPUs, FPGAs). Ευρέως διαδομένη είναι η ανάπτυξη παράλληλων δομών από Graphics Processing Units (GPU), τα οποία διαμορφώνονται σε clusters.

Σπουδαία αποτελέσματα με πληθώρα εφαρμογών στον τομέα της Μηχανικής Μάθησης προσφέρει η ανάπτυξη συστημάτων που περιλαμβάνουν εξειδικευμένο υλικό με τη χρήση FPGAs. Τρέχουσες τάσεις σε εργαλεία σχεδιασμού για FPGAs, όπως το High Level Synthesis, έχουν καταστήσει τον σχεδιασμό dedicated hardware accelerators πιο συμβατό με πρακτικές λογισμικού υψηλού επιπέδου (Software Engineering). Μέσω των τεχνικών αυτών μπορεί να επιτευχθεί η αποδοτικότερη "μεταφορά" των αλγορίθμων μηχανικής μάθησης στο επίπεδο του υλικού, τόσο από τη σκοπιά των απαιτούμενων πόρων υλοποίησης, όσο και από τη σκοπιά της βελτίωσης του ρυθμού επεξεργασίας.

Η αναγνώριση ψηφίων ή γραμμάτων είναι ένα κλασσικό παράδειγμα εφαρμογής των αλγορίθμων μηχανικής μάθησης και ειδικότερα των τεχνητών νευρωνικών δικτύων. Συγκεκριμένα η παρούσα διπλωματική εργασία αφορά στην υλοποίηση των νευρωνικών δικτύων τύπου Restricted Boltzmann Machines για την επιτάχυνση της διαδικασίας αναγνώρισης χειρόγραφων ψηφίων, με τη χρήση εξειδικευμένου υλικού και τελευταίας γενεάς FPGA - Xilinx Virtex 7.

Λέξεις κλειδιά: Αναγνώριση Χειρόγραφων Ψηφίων, Τεχνητή Νοημοσύνη, Τεχνητά Νευρωνικά Δίκτυα, Παράλληλη Επεξεργασία, Εξειδικευμένο Υλικό

Abstract

The rapid growth in data size and accessibility in recent years has as a result a new approach on system design philosophy with the use of algorithms of artificial intelligence.

These systems have the ability to learn and process vast amounts of data, which opens up new horizons in essential areas such as computational vision, pattern recognition, natural language processing, etc.

The most popular class of techniques that use artificial neural networks is called Deep Learning Networks and has a wide range of applications in everyday life as well as in industry. However, these models have very high demands on data storage and computing power, especially during the training process.

A solution to this complex design problem is attempted through the use of multiprocessing systems (Multiprocessors, GPUs, FPGAs). A widespread idea is the development of parallel architecture of Graphics Processing Units (GPU), which are formed in clusters.

Excellent results with a variety of applications in the field of Machine Learning are also provided by the development of systems that include specialized hardware using FPGAs. Current trends in design tools for FPGAs, such as High Level Synthesis, have made hardware accelerators more compliant with high-level software practices (Software Engineering). Through these techniques, it is possible to achieve more efficient porting of Machine Learning algorithms at the hardware level, both regarding the required implementation resources, but also the acceleration of the processing rate.

Recognizing digits or letters is a classic application example of the artificial neural network algorithms. Specifically, this Master Thesis deals with the implementation of Restricted Boltzmann Machines to accelerate the handwritten digits recognition process, using dedicated hardware and the latest generation FPGA - Xilinx Virtex 7.

Keywords: RBM, FPGA, Artificial Neural Networks, Pattern Recognition, Handwritten Digits Recognition, Hardware Accelerators, Cloud Services, HLS

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν με οποιονδήποτε τρόπο στην εκπόνηση αυτής της διπλωματικής εργασίας.

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου κ. Θεόδωρο Αντωνακόπουλο για τη συνεχή υποστήριξη και βοήθεια, για τις υποδείξεις του καθώς και την άριστη συνεργασία μας.

Τέλος, η εργασία αυτή αφιερώνεται στους γονείς μου, τον αδερφό μου Φώτη, τη γιαγιά μου Στέλλα και τη σύντροφό μου Ειρήνη τους οποίους ευχαριστώ διότι υπήρξαν πάντα στήριγμα για μένα και στους οποίους οφείλω όλη τη διαδρομή των σπουδών μου μέχρι σήμερα.

Πάτρα, Φεβρουάριος 2018

Νικόλαος Τουλγαρίδης

Περιεχόμενα

Περίληψη	5
Abstract	1
Ευχαριστίες	3
1 Εισαγωγή	8
1.1 Αντικείμενο της διπλωματικής	9
1.2 Δομή του Συγγράμματος	9
2 Παρουσίαση προβλήματος	11
2.1 Αναγνώριση Προτύπων και Μηχανική Μάθηση:	
Hardware Acceleration	11
2.1.1 Αναγνώριση Προτύπων: Εφαρμογές	12
2.1.2 Αναγνώριση ψηφίων-γραμμάτων:	
Παραδείγματα Δομών δεδομένων παρατήρησης	13
2.1.3 Αναγνώριση χειρόγραφων ψηφίων: Βάση Δεδομένων MNIST	15
3 Θεωρητική Προσέγγιση	16
3.1 Τεχνητή Νοημοσύνη	16
3.2 Νευρωνικά Δίκτυα	18
3.2.1 Βασικά στοιχεία λειτουργίας ενός νευρώνα	18
3.2.2 Αρχιτεκτονική Τεχνητών Νευρωνικών Δικτύων	19
3.3 Classification Restricted Boltzmann Machines	20
3.3.1 RBM Training και Classification	22
3.3.2 Generative Training Objective	23
3.3.3 Discriminative Training Objective	24
3.3.4 Hybrid Training Objective	25
3.4 Deep Learning - Deep Belief Networks	26
4 RBM στην Αναγνώριση Χειρόγραφων	
Ψηφίων	28
4.1 Μοντέλο Προσομοίωσης HDR-RBM	28
4.2 Ανάλυση παραμέτρων συστήματος	29
4.2.1 Πλήθος N_l κόμβων	29
4.2.2 Μέγεθος εικόνων - Πλήθος N_v κόμβων	29

4.2.3	Πλήθος Nh κόμβων	32
4.3	Επίδραση της τροποποίησης των MNIST Datasets στην ακρίβεια του συστήματος	33
4.4	Hardware Accelerator: Σχεδιαστικές επιλογές	35
4.4.1	Αριθμητική Υπολογισμών	35
5	Αρχιτεκτονική του Hardware Accelerator	38
5.1	Εξισώσεις Classification και Φάσεις υπολογισμού	38
5.2	Hardware Accelerator: Δομικό Διάγραμμα	39
5.2.1	Πρώτη Υπολογιστική Φάση	40
5.2.2	Δεύτερη Υπολογιστική Φάση	43
5.2.3	Τρίτη Υπολογιστική Φάση	46
5.3	Εφαρμογή τεχνικών βελτιστοποίησης	48
6	Προτυποποίηση Συστήματος και Μετρήσεις	52
6.1	Cloud Services	52
6.2	Πόροι συστήματος και Εργαλεία ανάπτυξης	53
6.3	Cloud Services: RBM Hardware Accelerator	54
6.3.1	Πρότυπα Διασύνδεσης	54
6.3.2	Native PCIe	55
6.3.3	CAPI	58
6.4	Συγκριτικές μετρήσεις για τις διαφορετικές υλοποιήσεις του Cloud Server: HDR-RBM Hardware Accelerator	62
7	Επίλογος	64
7.1	Συμπεράσματα	64
7.2	Μελλοντικές επεκτάσεις	64
	Βιβλιογραφία	68

Κατάλογος Σχημάτων

1.1	Machine Learning	8
2.1	Αναγνώριση κινούμενων οχημάτων	11
2.2	Εφαρμογές αναγνώριση ψηφίων	12
2.3	Παραδείγματα αναγνώρισης ψηφίων με χρήση στη Βιομηχανία	12
2.4	Παραδείγματα αναγνώρισης οργάνων με εφαρμογή στον κλάδο της Ιατρικής Επιστήμης	13
2.5	MNIST Dataset	14
2.6	The Street View House Numbers και The Chars74K datasets	14
3.1	Reinforcement Learning	17
3.2	Clustering	17
3.3	Διάγραμμα Νευρώνα εγκεφάλου	18
3.4	Νευρώνας και Νευρωνικά Δίκτυα	19
3.5	Γενική Αρχιτεκτονική Νευρωνικού Δικτύου και stacked RBM	20
3.6	Αρχιτεκτονική δικτύου RBM	21
3.7	Μοντελοποίηση της από κοινού κατανομής των Visible και Target nodes με χρήση Discriminative RBM	22
3.8	Αλγόριθμος $k - step$ Contrastive Divergence	24
3.9	Δομή δικτύου DBN	26
3.10	Δίκτυο DBN 3 Hidden Layers	26
4.1	Εφαρμογή HDR σε δίκτυο RBM	29
4.2	MNIST Binarization technique	30
4.3	Bicubic Interpolation	31
4.4	Μείωση μεγέθους εικόνων: από 784 σε 256 pixels	31
4.5	Ακρίβεια RBM vs N_h	32
4.6	Η επίδραση των διαφορετικών training datasets στην ακρίβεια του Classification	35
4.7	RBM Accuracy vs Number of Fractional Bits	36
5.1	Αρχιτεκτονική RBM Hardware Accelerator	39
5.2	Αρχιτεκτονική 1 ^{ης} Υπολογιστικής Φάσης	40
5.3	Υπολογιστική Φάση 1: Χρονικό Διάγραμμα	41
5.4	Υπολογιστική Φάση 1: Timing Waveform	42
5.5	Αρχιτεκτονική 2 ^{ης} Υπολογιστικής Φάσης	43
5.6	Υπολογιστική Φάση 2: Εσωτερική Αρχιτεκτονική	43

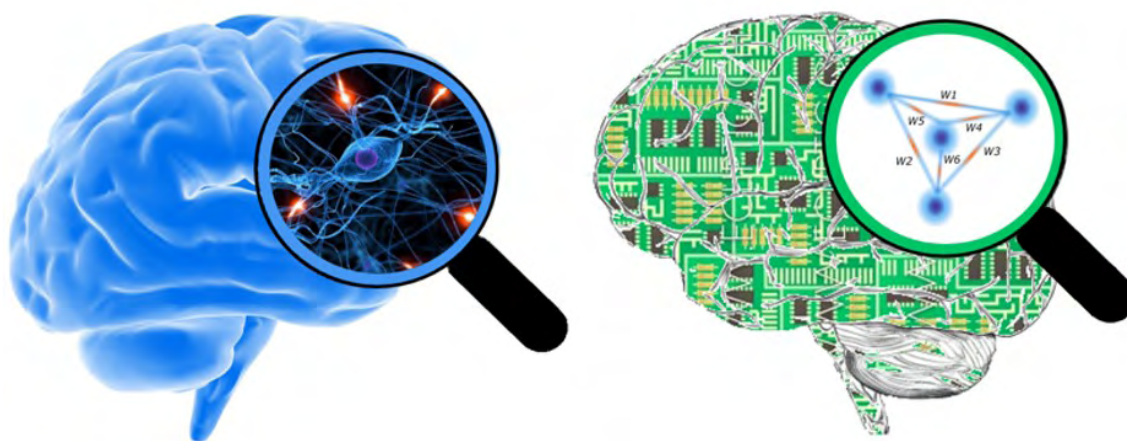
5.7	Υπολογιστική Φάση 2: Χρονικό Διάγραμμα	44
5.8	Υπολογιστική Φάση 2: Timing Waveform	45
5.9	Αρχιτεκτονική 3 ^{ης} Υπολογιστικής Φάσης	46
5.10	Υπολογιστική Φάση 3: Αρχιτεκτονική 1 ^{ου} κυκλώματος	47
5.11	Υπολογιστική Φάση 3: Αρχιτεκτονική 2 ^{ου} κυκλώματος	47
5.12	Υπολογιστική Φάση 3: Timing Waveform	48
5.13	Συνδυασμός Υπολογιστικών Φάσεων και τελική αρχιτεκτονική Hardware Accelerator	49
5.14	Pipeline Stages εξωτερικά των υπολογιστικών φάσεων	49
5.15	Accelerator Timing Waveform	50
6.1	Cloud Services: Handwritten Digits Recognition	53
6.2	Power8 Server - Alpha Data Board	54
6.3	Αρχιτεκτονική συστήματος Native PCIe	55
6.4	Αρχιτεκτονική Native PCIe Hardware Block Design	56
6.5	Native PCIe FPGA Resource Utilization	57
6.6	CAPI Hardware Acceleration	58
6.8	CAPI Hardware Ecosystem	58
6.7	CAPI Solution Flow	59
6.9	Αρχιτεκτονική συστήματος CAPI	60
6.10	Αρχιτεκτονική CAPI Hardware Block Design: Accelerator Functional Unit	61
6.11	Αρχιτεκτονική CAPI Hardware Block Design: Neural Networks CAPI Adaptation Layer	61
6.12	Native PCIe System Acceleration	62
6.13	Native PCIe and CAPI System Acceleration	63

Κατάλογος Πινάκων

2.1	Αλγόριθμοι αναγνώρισης χειρόγραφων ψηφίων και ακρίβεια	15
4.1	Ακρίβεια και Πολυπλοκότητα σε συνάρτηση με N_v και N_h	33
5.1	Απαιτούμενοι πόροι πολλαπλασιασμού - πολύπλεξης (Virtex - 7)	41
5.2	Συγκριτικός πίνακας απαιτούμενων πόρων DSP48E και LUT για τη φάση 2 (Target device: Virtex 7, xc7vx690tffg1157-2)	46
5.3	Throughput συνολικού συστήματος με μεταβλητό αριθμό NNs	50
5.4	Hardware Accelerator: Resource Utilization, (Target device: Virtex 7, xc7vx690tffg1157-2)	51
6.1	CAPI FPGA Resource Utilization	60

Κεφάλαιο 1

Εισαγωγή



Σχήμα 1.1: *Machine Learning*

Ζούμε στην εποχή των *Big Data*, όπου τεράστιες ποσότητες δομημένων, ημιδομημένων και αδόμητων δεδομένων δημιουργούνται με συνεχώς αυξανόμενο ρυθμό. Από την επεξεργασία και την ανάλυση των δεδομένων αυτών μπορούν να προκύψουν πολύτιμες πληροφορίες. Η πιο συνηθισμένη προσέγγιση για τη συλλογή αυτού του είδους δεδομένων είναι η ομαδοποίησή τους σε μεγάλα σύνολα δεδομένων με την προσθήκη κάποιας πληροφορίας γνωστής ως *metadata*. Στη συνέχεια, είναι δυνατή η εφαρμογή αλγορίθμων τεχνητής νοημοσύνης και συγκεκριμένα μηχανικής μάθησης προκειμένου να εντοπιστούν τα επαναλαμβανόμενα πρότυπα. Τα τεχνητά νευρωνικά δίκτυα είναι μια ταχέως αναπτυσσόμενη κατηγορία δομών μηχανικής μάθησης που δίνουν στους υπολογιστές τη δυνατότητα να μάθουν να εκτελούν συγκεκριμένες εργασίες χωρίς να έχουν προγραμματιστεί ρητά για αυτόν το σκοπό. Αποτελούνται από διαφορετικά επίπεδα για την ανάλυση και την εκμάθηση πάνω σε ένα σύνολο δεδομένων που αποτελούν τα πρότυπα εκπαίδευσης. Κάθε επίπεδο αποτελείται από ένα μεγάλο αριθμό διασυνδεδεμένων στοιχείων επεξεργασίας (νευρώνες), που εργάζονται μαζί για να μάθουν από υπάρχοντα δεδομένα με στόχο να επιλύσουν συγκεκριμένα προβλήματα, λαμβάνοντας σωστές αποφάσεις.

Οι μηχανές Boltzmann (BM) είναι ένα χαρακτηριστικό παράδειγμα δομής νευρωνικού δικτύου. Γενικά, η εκμάθηση μιας μηχανής Boltzmann παρουσιάζει σημαντικές υπολογιστικές

απαιτήσεις. Ωστόσο, το πρόβλημα της εκπαίδευσης του δικτύου μπορεί να απλοποιηθεί με την επιβολή περιορισμών στην τοπολογία του δικτύου, γεγονός που οδηγεί στα *Restricted Boltzmann Machines*. Τα RBM είναι δομημένα ως διμερή μη κατευθυνόμενα γραφήματα, τα οποία επιτρέπουν την αποδοτική εξαγωγή συμπερασμάτων και ιδιαίτερα σύνθετων χαρακτηριστικών. Τα τελευταία χρόνια, πολλές εφαρμογές βασισμένες σε RBM έχουν αναπτυχθεί για μια μεγάλη ποικιλία μαθησιακών προβλημάτων, όπως η ταξινόμηση των εικόνων, η αναγνώριση ομιλίας κ.ο.κ. Ένα τέτοιο παράδειγμα εφαρμογής είναι η αναγνώριση χειρόγραφων ψηφίων. Οι πόροι αποθήκευσης και ο απαιτούμενος χρόνος όχι μόνο για την εκπαίδευση ενός RBM, αλλά και για την πρόβλεψη σε πραγματικό χρόνο, αυξάνονται εκθετικά με τον αριθμό των παραμέτρων. Αυτό καθιστά την ανάπτυξη μιας πλήρους εφαρμογής αναγνώρισης χειρόγραφων ψηφίων αρκετά σύνθετη, ειδικά για τις περιπτώσεις όπου αξιοποιούνται οι πόροι ενός απλού επεξεργαστή γενικού σκοπού. Έτσι παρουσιάζει ιδιαίτερο ενδιαφέρον η ανάπτυξη μεθόδων για την από κοινού επεξεργασία των δεδομένων με την παράλληλη λειτουργία τόσο επεξεργαστών όσο και μονάδων εξειδικευμένης επεξεργασίας και επιτάχυνσης με τη χρήση FPGAs. Υπάρχουν πολλά παραδείγματα αξιοποίησης ολοκληρωμένων συστημάτων επεξεργασίας, ιδιαίτερα σε εφαρμογές παροχής υπηρεσιών, όπως για παράδειγμα τα συστήματα *Cloud Services*.

1.1 Αντικείμενο της διπλωματικής

Η παρούσα διπλωματική εργασία επικεντρώθηκε στην ανάλυση των αλγορίθμων των RBM νευρωνικών δικτύων και στη σχεδίαση και υλοποίηση ενός συστήματος αναγνώρισης χειρόγραφων ψηφίων. Στόχος ήταν η ανάπτυξη, σε επίπεδο υλικού, μίας βιβλιοθήκης που θα προσφέρει υψηλή προσαρμοστικότητα αλλά και ακρίβεια και επεξεργαστική ισχύ για την επίλυση του συγκεκριμένου προβλήματος. Στα πλαίσια της εργασίας υλοποιήθηκε ένας *Custom Hardware Accelerator* και ενσωματώθηκε σε σύστημα που παρέχει υπηρεσίες αναγνώρισης ψηφίων σε μεγάλο όγκο δεδομένων. Η παρούσα μελέτη θα εστιάσει και θα περιγράψει όλα τα στάδια που ακολουθήθηκαν για την επιτυχή διεκπεραίωση της διπλωματικής εργασίας, ενώ προς το τέλος θα εξηγηθούν εκτενώς τα αποτελέσματά της.

1.2 Δομή του Συγγράμματος

Η εργασία αυτή είναι οργανωμένη σε έξι κεφάλαια:

Στο 2^ο κεφάλαιο παρουσιάζονται ορισμένα ευρέως διαδεδομένα προβλήματα αναγνώρισης προτύπων, καθώς και δομές δεδομένων που περιλαμβάνουν εικόνες από δείγματα, όπως τα χειρόγραφα ψηφία.

Το 3^ο κεφάλαιο αφιερώνεται στην παρουσίαση του βασικού θεωρητικού υποβάθρου στο οποίο βασίστηκε η εργασία.

Στο 4^ο κεφάλαιο αναλύεται η μέθοδος με την οποία συνδυάστηκαν το πρόβλημα της αναγνώρισης χειρόγραφων ψηφίων με τα νευρωνικά δίκτυα RBM. Ακόμα στο κεφάλαιο

αυτό περιγράφεται η μοντελοποίηση του συστήματος που προηγήθηκε της υλοποίηση του RBM Hardware Accelerator.

Στο 5^ο κεφάλαιο περιγράφεται η αρχιτεκτονική του RBM Hardware Accelerator, για το πρόβλημα της αυτόματης αναγνώρισης χειρόγραφων ψηφίων.

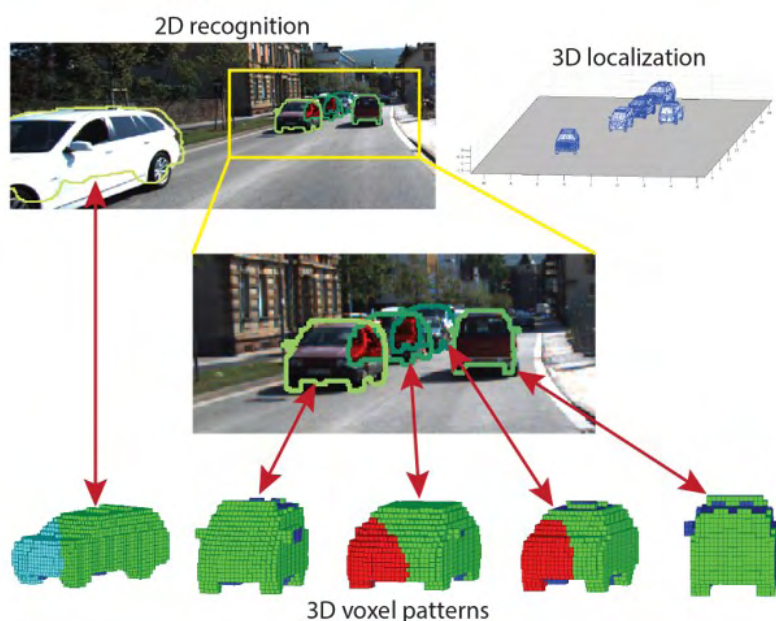
Το 6^ο κεφάλαιο είναι αφιερωμένο στην προτυποποίηση του συνολικού συστήματος. Συγκεκριμένα περιγράφονται οι μέθοδοι ενσωμάτωσης του RBM Hardware Accelerator σε έναν server, ενώ παρουσιάζονται μετρήσεις που αφορούν την απόδοση του συστήματος.

Τέλος, το 7^ο κεφάλαιο παραθέτει τα συμπεράσματα από το συνολικό έργο που διεξήχθει στο πλαίσιο της διπλωματικής εργασίας καθώς και ορισμένες σκέψεις για μελλοντικές επεκτάσεις του συστήματος.

Κεφάλαιο 2

Παρουσίαση προβλήματος

2.1 Αναγνώριση Προτύπων και Μηχανική Μάθηση: Hardware Acceleration



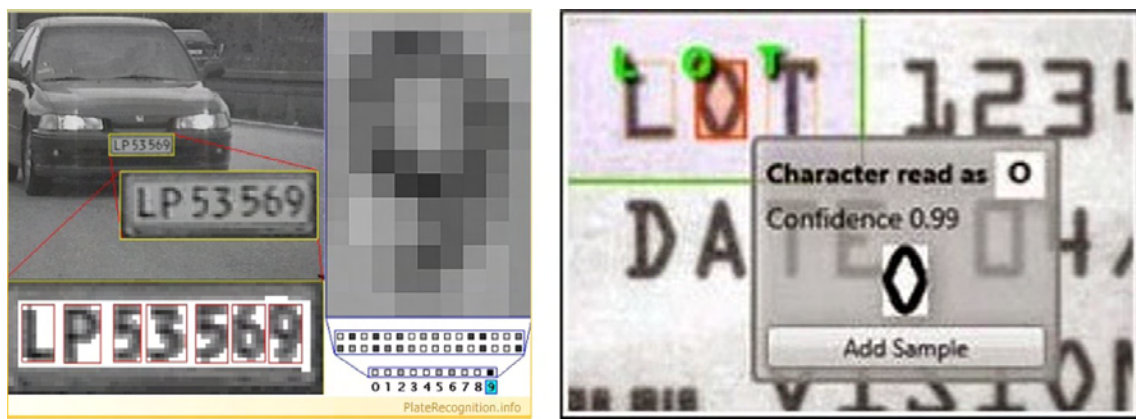
Σχήμα 2.1: Αναγνώριση κινούμενων οχημάτων

Η αναγνώριση προτύπων είναι ένας ώριμος, αλλά συναρπαστικός και ταχέως αναπτυσσόμενος τομέας, ο οποίος κρύβεται πίσω από τις σύγχρονες εξελίξεις σε γνωστικά πεδία όπως η υπολογιστική όραση, η επεξεργασία εικόνων, η ανάλυση κειμένων και εγγράφων και τα νευρωνικά δίκτυα. Είναι τομέας της μηχανικής μάθησης και βρίσκει, επίσης, εφαρμογές σε περιοχές που αναπτύσσονται με ραγδαίους ρυθμούς τα τελευταία χρόνια. Ως ένας γενικός ορισμός της αναγνώρισης προτύπων μπορεί να δοθεί ο εξής: *Κλάδος της τεχνητής νοημοσύνης που ασχολείται με την αυτοματοποιημένη ανάλυση σε συλλογές χαρακτηριστικών από αντικείμενα, γεγονότα κ.ο.κ. με σκοπό την ταξινόμησή τους σε κατηγορίες.*

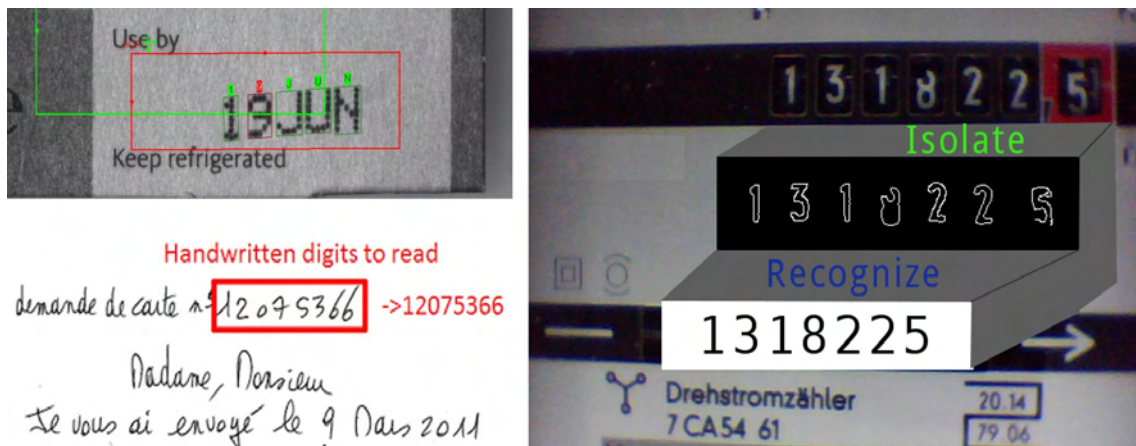
Στις σημερινές ανάγκες υπολογιστικής ισχύος και στην εποχή της παράλληλης κατα-

νεμημένης επεξεργασίας, μία σύγχρονη απαίτηση είναι η ταχεία εκτέλεση μιας πληθώρας αλγορίθμων με τη χρήση εξειδικευμένων μονάδων υπολογισμών. Η διαδικασία αυτή παρουσιάζει τόσο ακαδημαϊκό όσο και βιομηχανικό ενδιαφέρον. Κρίθηκε, λοιπόν, σκόπιμο η παρούσα διπλωματική εργασία να ερευνήσει και να συνδυάσει τομείς της Μηχανικής Μάθησης και της χρήσης εξειδικευμένων Hardware Accelerators. Στόχος ήταν η ανάλυση, σχεδίαση και υλοποίηση μιας βιβλιοθήκης που περιλαμβάνει έναν Hardware Accelerator ικανό να βελτιώσει την ταχύτητα εκτέλεσης για ένα πρόβλημα αναγνώρισης προτύπων και συγκεκριμένα της αυτόματης αναγνώρισης ψηφίων.

2.1.1 Αναγνώριση Προτύπων: Εφαρμογές



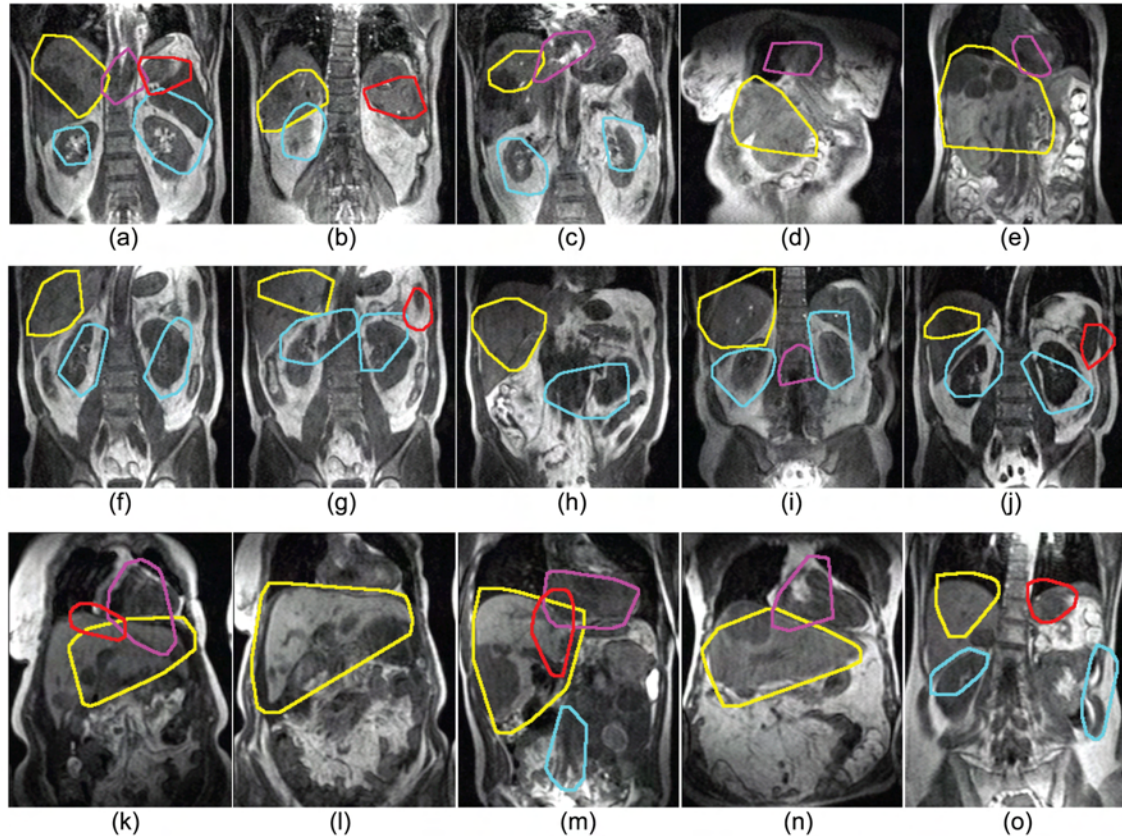
Σχήμα 2.2: Εφαρμογές αναγνώριση ψηφίων



Σχήμα 2.3: Παραδείγματα αναγνώρισης ψηφίων με χρήση στη Βιομηχανία

Η αναγνώριση προτύπων βρίσκει ευρεία εφαρμογή στην αυτοματοποίηση διαδικασιών ταξινόμησης και ομαδοποίησης σε πληθώρα προβλημάτων της καθημερινής ζωής. Η αυτοματοποίηση αυτή μπορεί να επιταχύνει και να απλοποιήσει χρονοβόρες διαδικασίες, προσφέροντας, πολλές φορές, αναβάθμιση του επιπέδου των υπηρεσιών που παρέχονται στο σύνολο της κοινωνίας. Επιπρόσθετα, ιδιαίτερα σημαντική είναι η συνεισφορά των αλγορίθμων αναγνώρισης προτύπων σε ευαίσθητους κλάδους όπως η υγεία, η εκπαίδευση αλλά ακόμα και η καταπολέμηση του εγκλήματος. Στις εικόνες 2.1, 2.2 και 2.3 δίνονται ορισμένα

από τα παραδείγματα εφαρμογής της αναγνώρισης προτύπων και ιδιαίτερα της αναγνώρισης ψηφίων ή γραμμμάτων. Μέσα από τα συγκεκριμένα παραδείγματα αλλά και από πολλά ακόμα που συναντά κανείς στην καθημερινή ζωή, αλλά και στη βιομηχανία, γίνεται κατανοητή η σημασία της επένδυσης πόρων και επιστημονικής έρευνας σε αυτούς τους τομείς.



Σχήμα 2.4: Παραδείγματα αναγνώρισης οργάνων με εφαρμογή στον κλάδο της Ιατρικής Επιστήμης

Ένα πολύ χαρακτηριστικό παράδειγμα εφαρμογής των τεχνικών της αναγνώρισης προτύπων είναι η αυτόματη αναγνώριση φυσιολογιών ή παθολογιών στο ανθρώπινο σώμα. Για το σκοπό αυτό, ως δεδομένα εκπαίδευσης αξιοποιούνται σύνολα από εικόνες που έχουν προέρθει από μαγνητική ή αξονική τομογραφία. Στην εικόνα 2.4 φαίνονται τα αποτελέσματα ενός αλγορίθμου μηχανικής μάθησης που είναι υπεύθυνος για την αναγνώριση των ζωτικών οργάνων σε ένα ανθρώπινο σώμα.

2.1.2 Αναγνώριση ψηφίων-γραμμμάτων: Παραδείγματα Δομών δεδομένων παρατήρησης

Τα τελευταία χρόνια έχει δοθεί ιδιαίτερη έμφαση, κυρίως από την επιστημονική κοινότητα, στη συλλογή και την προτυποποίηση μίας σειράς δεδομένων, τα οποία μπορούν να αξιοποιηθούν ως στοιχεία εισόδου σε συστήματα αναγνώρισης προτύπων και επεξεργασίας εικόνων. Τα δεδομένα αυτά ποικίλουν τόσο σε μορφή όσο και σε περιεχόμενο καθώς υπάρχει υψηλή εξάρτηση από την εφαρμογή που πρόκειται να αξιοποιηθούν. Συγκεκριμένα για συστήματα που προσπαθούν να επιλύσουν προβλήματα που αφορούν αναγνώριση προτύπων



Σχήμα 2.5: *MNIST Dataset*



Σχήμα 2.6: *The Street View House Numbers και The Chars74K datasets*

σε εικόνες, συνήθως συλλέγονται δεδομένα που απεικονίζουν στοιχεία της καθημερινότητας. Παρακάτω θα δοθούν τρία χαρακτηριστικά παραδείγματα δομών δεδομένων ([1]) που βρίσκουν εφαρμογή σε αλγορίθμους αναγνώρισης ψηφίων ή γραμμάτων.

1. Βάση δεδομένων χειρόγραφων ψηφίων *MNIST Dataset* [2], εικόνα 2.5.
2. The Street View House Numbers (*SVHN*) dataset [3],[4], εικόνα 2.6. Η συλλογή αυτή περιλαμβάνει εικόνες από αριθμούς σπιτιών και εμφανίζει ιδιαίτερη διασπορά ως προς τα χαρακτηριστικά των εικόνων αυτών, για παράδειγμα το χρώμα ή το σχήμα των ψηφίων.
3. The Chars74K dataset, το οποίο περιλαμβάνει στοιχεία από τυποποιημένα γράμματα και ψηφία [5], εικόνα 2.6. Η συλλογή αυτή αποτελείται από λατινικούς χαρακτήρες και από ψηφία. Επίσης, αυτή η βιβλιοθήκη εμφανίζει υψηλή διασπορά στα χαρακτηριστικά των εικόνων της.

2.1.3 Αναγνώριση χειρόγραφων ψηφίων: Βάση Δεδομένων MNIST

Από τη στιγμή που η παρούσα διπλωματική ασχολήθηκε πιο ειδικά με την αυτόματη αναγνώριση χειρόγραφων ψηφίων, κρίθηκε ως πιο κατάλληλη για χρήση η βιβλιοθήκη χειρόγραφων ψηφίων MNIST. Η βάση δεδομένων MNIST είναι ένα σύνολο εικόνων που αναπτύχθηκε από τους *Yann LeCun* και *Christopher Burges* για την αξιολόγηση των μοντέλων μηχανικής μάθησης στο πρόβλημα της ταξινόμησης χειρόγραφων ψηφίων (*Handwritten Digits Recognition*).

Από τον όρο *χειρόγραφα ψηφία* γίνεται κατανοητό ότι η διαδικασία της αναγνώρισης περιλαμβάνει τη λήψη της απόφασης για το ποιο ψηφίο, από το 0 μέχρι το 9, κωδικοποιεί η κάθε εικόνα. Το σύνολο δεδομένων κατασκευάστηκε από σαρωμένα έγγραφα που διατίθενται από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (*National Institute of Standards and Technology - NIST*) των Ηνωμένων Πολιτειών. Από αυτό προέρχεται το όνομα της βάσης δεδομένων χειρόγραφων ψηφίων, ως τροποποιημένο *NIST* ή *Modified NIST*. Κάθε εικόνα αποτελείται από 28x28 pixels (σύνολο 784). Η βάση χωρίζεται σε δύο σύνολα δειγμάτων. Ένα σύνολο από 60.000 εικόνες είναι αφιερωμένο στις διαδικασίες της *εκμάθησης (training)* του αλγορίθμου που προσπαθεί αναγνωρίσει τα πρότυπα στις εικόνες αυτές. Το δεύτερο σύνολο περιλαμβάνει 10.000 εικόνες που χρησιμοποιούνται για τη διαδικασία της *ταξινόμησης (classification)*, η οποία ακολουθεί αυτή της εκμάθησης σε έναν αλγόριθμο αναγνώρισης προτύπων. Επιπρόσθετα υπάρχει και μία μεγαλύτερου μεγέθους βιβλιοθήκη που περιλαμβάνει 8.1 εκατομμύρια εικόνες. Ωστόσο για τους σκοπούς της διπλωματικής εργασίας η χρήση των δύο μικρών datasets κρίθηκε επαρκής.

Στον πίνακα 2.1 παρατίθενται ορισμένοι από τους πιο δημοφιλείς αλγορίθμους ([6]) που αξιοποιούν τη MNIST Database για την επίλυση του προβλήματος της αναγνώρισης χειρόγραφων ψηφίων.

<i>Algorithm</i>	<i>Accuracy</i>
Deep Boltzmann Machines, AISTATS 2009	91%
Convolutional Kernel Networks, arXiv 2014	96%
Competitive Multi-scale Convolution , arXiv 2015	97%
DropConnect, ICML 2013	98%

Πίνακας 2.1: Αλγόριθμοι αναγνώρισης χειρόγραφων ψηφίων και ακρίβεια

Κεφάλαιο 3

Θεωρητική Προσέγγιση

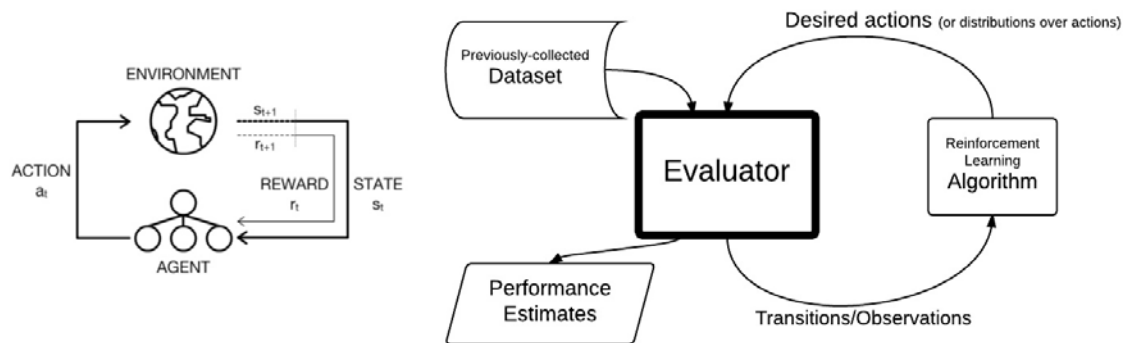
Το 1959 ο Arthur Samuel περιέγραψε τον επιστημονικό τομέα της Μηχανικής Μάθησης ως το “πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μάθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί”.

3.1 Τεχνητή Νοημοσύνη

Η Τεχνητή Νοημοσύνη είναι ένας τομέας της επιστήμης των υπολογιστών που δίνει έμφαση στη δημιουργία “έξυπνων” μηχανών που λειτουργούν και αντιδρούν αντιγράφοντας σε ένα βαθμό τις διεργασίες του ανθρώπινου εγκεφάλου. Υπάρχουν όμως πολλές διαδικασίες τις οποίες, ενώ τα ζώα και οι άνθρωποι είναι σε θέση να διεκπαιρώσουν μάλλον εύκολα, παραμένουν ακόμα και στις αρχές του 21ου αιώνα, μακριά από τις άμεσες δυνατότητες των υπολογιστών. Η έννοια της νοημοσύνης μπορεί να οριστεί με πολλούς τρόπους. Στο παρών σύγγραμμα ορίζεται ως *η ικανότητα λήψης σωστών αποφάσεων, σύμφωνα με κάποιο κριτήριο*. Για να ληφθούν οι βέλτιστες αποφάσεις απαιτείται η γνώση, σε μορφή που να είναι λειτουργική, δηλαδή, να μπορεί να χρησιμοποιηθεί για την ερμηνεία αισθητηριακών δεδομένων και τη χρήση αυτών των πληροφοριών για τη λήψη αποφάσεων.

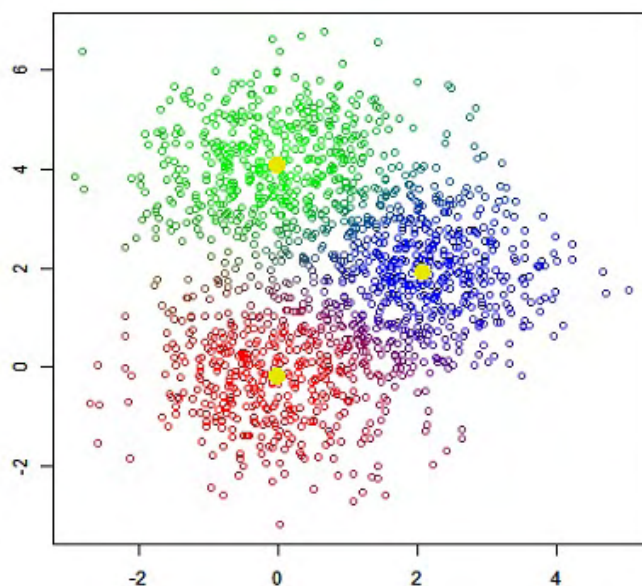
Η τεχνολογία της γνώσης αποτελεί βασικό μέρος της έρευνας γύρω από την Τεχνητή Νοημοσύνη. Οι μηχανές μπορούν συχνά να ενεργούν όπως οι άνθρωποι, μόνο αν έχουν *ικανό αριθμό από πληροφορίες* σχετικές με το πρόβλημα που προσπαθούν να λύσουν. Οι αλγόριθμοι της τεχνητής νοημοσύνης απαιτούν πρόσβαση σε αντικείμενα, κατηγορίες, ιδιότητες και σχέσεις για την εφαρμογή της γνωστικής διαδικασίας.

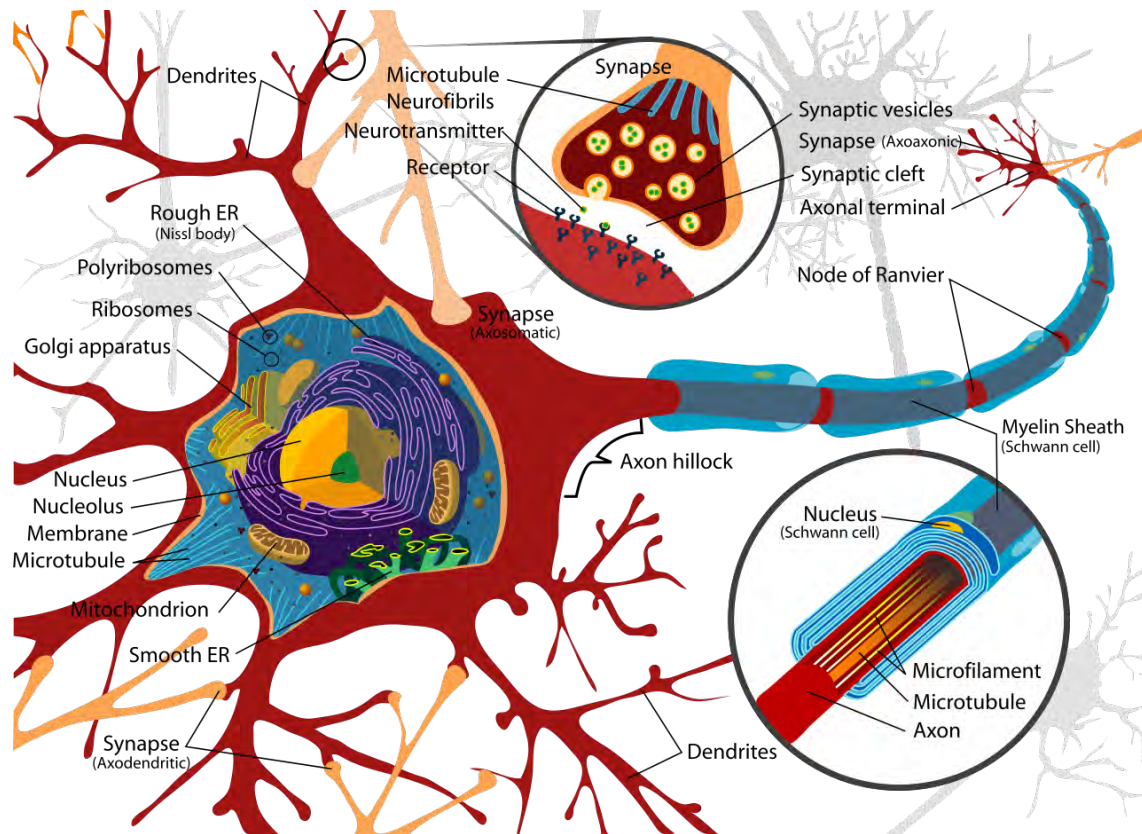
Η Μηχανική Μάθηση είναι επίσης ένας βασικός κλάδος της Τεχνητής Νοημοσύνης [7]. Η μάθηση χωρίς καμία εποπτεία (unsupervised learning) απαιτεί την ικανότητα ταυτοποίησης προτύπων σε ροές δεδομένων, ενώ η μάθηση με επαρκή εποπτεία (supervised learning) περιλαμβάνει τις διαδικασίες της ταξινόμησης (classification) και της αριθμητικής παλινδρόμησης (numerical regression). Η ταξινόμηση μπορεί να καθορίσει την κατηγορία στην οποία ανήκει ένα αντικείμενο, ενώ η παλινδρόμηση ασχολείται με τη συλλογή συνόλων αριθμητικών παραδειγμάτων εισόδου και εξόδου, με στόχο την εύρεση μίας συνάρτησης η οποία καλύτερα περιγράφει την παραγωγή κατάλληλων εξόδων από τις αντίστοιχες εισόδους. Η

Σχήμα 3.1: *Reinforcement Learning*

μαθηματική ανάλυση των αλγορίθμων μηχανικής μάθησης και η απόδοσή τους είναι ένας καλά καθορισμένος κλάδος της θεωρητικής πληροφορικής που συχνά αναφέρεται ως *θεωρία υπολογιστικής μάθησης*.

Είναι δύσκολο να αναπαραχθεί η ανθρώπινη διαίσθηση σε μια μηχανή, κυρίως επειδή οι άνθρωποι συχνά μαθαίνουν και εκτελούν αποφάσεις με ενστικτώδη τρόπο. Όπως και τα παιδιά έτσι και οι μηχανές απαιτούν μια εκτεταμένη περίοδο *εκμάθησης (training)*, η οποία βασίζεται σε παραδείγματα εκπαίδευσης, με στόχο την υπαγόρευση της μελλοντικής συμπεριφοράς. Οι τεχνικές εκμάθησης μεταξύ άλλων περιλαμβάνουν τεχνικές όπως: η εκμάθηση ρόλων (role learning), η προσαρμογή παραμέτρων (parameter adjustment), η εκσκαφή (chunking), η ομαδοποίηση (clustering), η διόρθωση σφαλμάτων (mistake correction), η τεχνική back propagation και η εξαναγκαστική μάθηση (reinforcement learning).

Σχήμα 3.2: *Clustering*



Σχήμα 3.3: Διάγραμμα Νευρώνα εγκεφάλου

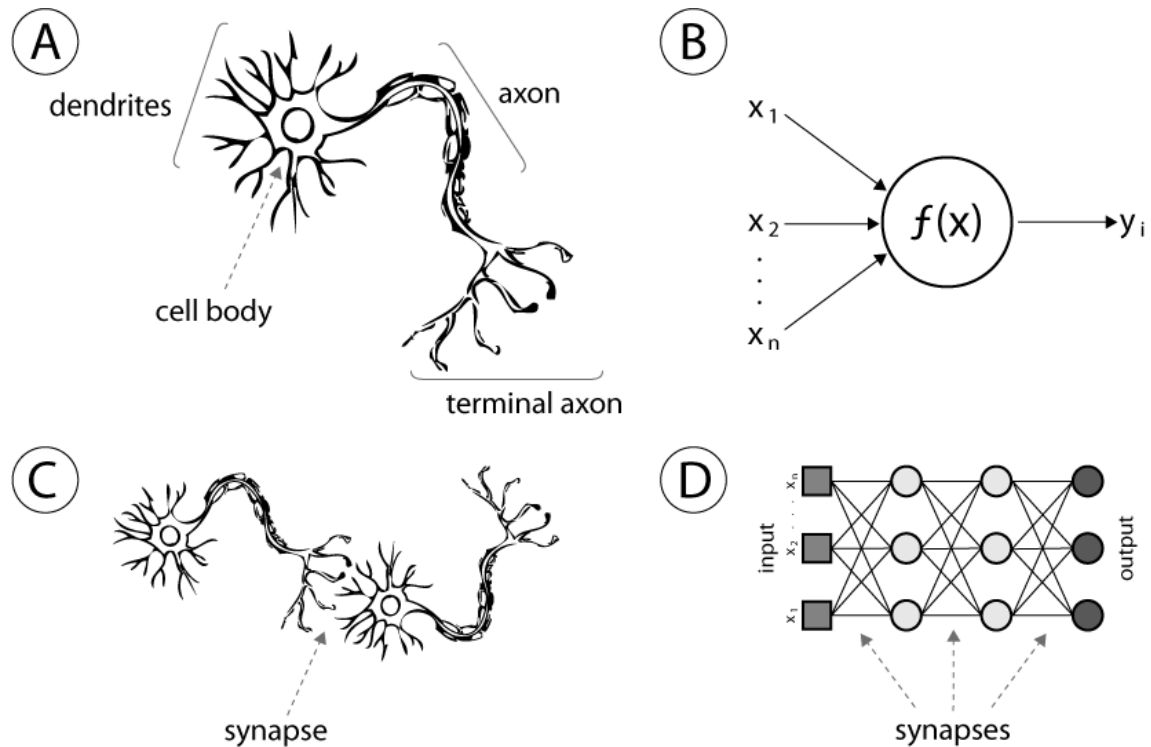
3.2 Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα είναι κλάδος της τεχνητής νοημοσύνης. Το όνομά τους προέρχεται από τα δομικά κύτταρα του εγκεφάλου, τους νευρώνες, λόγω του ότι προσπαθούν να αξιοποιήσουν τον τρόπο που αυτοί λειτουργούν. Η διαφορά μεταξύ των τεχνητών νευρωνικών δικτύων και των κλασσικών νευρωνικών δικτύων είναι ότι η λέξη τεχνητή προστέθηκε στη φράση έτσι ώστε οι άνθρωποι της υπολογιστικής νευροβιολογίας να μπορούν ακόμα να χρησιμοποιούν τον όρο νευρωνικό δίκτυο για να αναφερθούν στο έργο τους. Ένα Δίκτυο Τεχνητών Νευρώνων (Artificial Neural Network), είναι ένα υπολογιστικό μοντέλο βασισμένο στη δομή των βιολογικών νευρωνικών δικτύων. Είναι σαν ένα τεχνητό νευρικό σύστημα για τη λήψη, επεξεργασία και μετάδοση πληροφοριών από την σκοπιά της Πληροφορικής επιστήμης.

3.2.1 Βασικά στοιχεία λειτουργίας ενός νευρώνα

Το κύριο δομικό στοιχείο του νευρικού συστήματος γενικά, και ειδικότερα του εγκεφάλου, είναι ο νευρώνας ή το νευρικό κύτταρο. Ένας νευρώνας είναι ένα ηλεκτρικά διεγέρσιμο κύτταρο που επεξεργάζεται και μεταδίδει πληροφορίες με ηλεκτροχημική σηματοδότηση. Σε αντίθεση με άλλα κύτταρα, οι νευρώνες δεν μπορούν να αντικατασταθούν από νέους μετά την απώλειά τους.

Ο μέσος ανθρώπινος εγκέφαλος έχει περίπου 100 δισεκατομμύρια νευρώνες (ή νευρικά



Σχήμα 3.4: Νευρώνας και Νευρωνικά Δίκτυα

κύτταρα) και πολλά άλλα βοηθητικά (γνωστά ως νευρογλοιακά κύτταρα) που χρησιμεύουν για την υποστήριξη και την προστασία των νευρώνων. Κάθε νευρώνας μπορεί να συνδεθεί με έως και 10.000 άλλους νευρώνες, μεταφέροντας σήματα ο ένας στον άλλον μέσω περισσότερων από 1.000 τρισεκατομμυρίων συναπτικών συνδέσεων. Οι συνάψεις αυτές ισοδυναμούν, σύμφωνα με κάποιες εκτιμήσεις, με έναν υπολογιστή επεξεργαστικής ισχύος 1 τρισεκατομμυρίου δυαδικών ψηφίων ανά δευτερόλεπτο.

Ένας τυπικός νευρώνας διαθέτει ένα *σώμα* (το βολβώδες κυτταρικό σώμα που περιέχει τον πυρήνα του κυττάρου), τους *δενδρίτες* (μακριές, φτερωτές ίνες που συνδέονται με το κυτταρικό σώμα σε ένα σύνθετο διακλαδιστικό «δενδριτικό δέντρο») και έναν *νευράξονα* (ένα ειδικό, διακλαδισμένο κυτταρικό νήμα). Κάθε νευρώνας διατηρεί μια μεταβαλλόμενη *τάση* κατά μήκος της μεμβράνης του, λόγω των μεταβολικά οδηγούμενων διαφορών στα ιόντα νατρίου, καλίου, χλωρίου και ασβεστίου εντός του κυττάρου. Καθένα από αυτά παρουσιάζει διαφορετικό φορτίο. Αν η τάση αλλάξει σημαντικά, δημιουργείται ένας ηλεκτροχημικός παλμός που ονομάζεται *δυναμικό ενέργειας* (ή *νευρικός παλμός*). Αυτή η ηλεκτρική δραστηριότητα μπορεί να μετρηθεί και να εμφανιστεί ως μορφή κύματος που ονομάζεται *εγκεφαλικός ρυθμός*. Τα παραπάνω περιγράφονται από το σχήμα 3.3.

3.2.2 Αρχιτεκτονική Τεχνητών Νευρωνικών Δικτύων

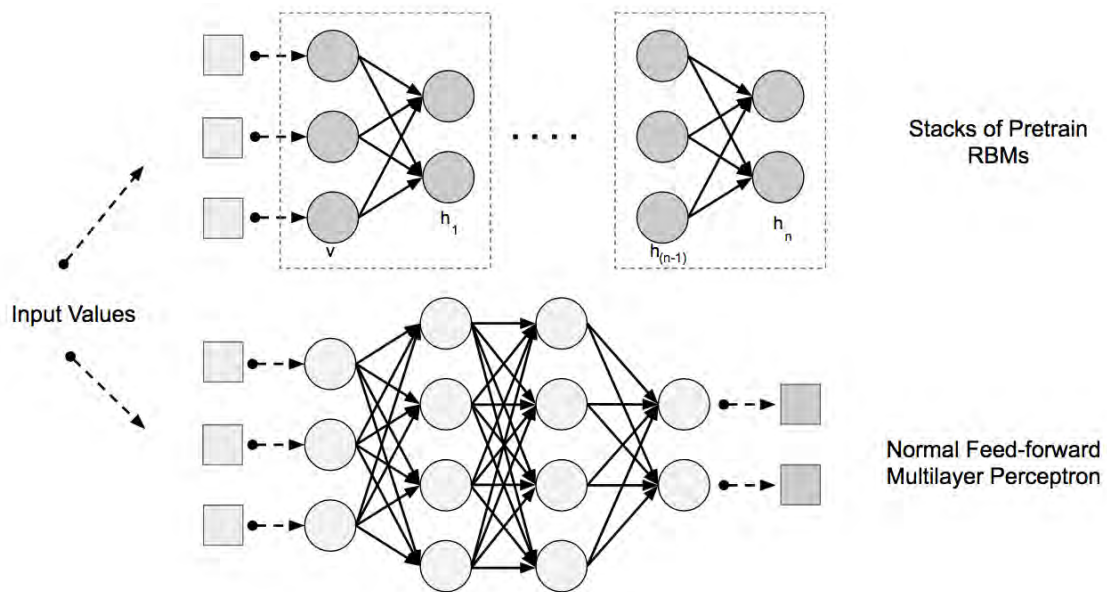
Ένας θεμελιώδης όρος για την κατανόηση των τεχνητών νευρωνικών δικτύων αποτελεί το *perceptron*. Το perceptron είναι ένας *δυαδικός ταξινομητής*. Με άλλα λόγια είναι μία συνάρτηση η οποία αντιστοιχίζει τις εισόδους x (ένα διάνυσμα τιμών) με την έξοδο $f(x)$ (μία δυαδική τιμή). Τα τεχνητά νευρωνικά δίκτυα μπορούν να χαρακτηριστούν και ως πολυε-

πίπεδα perceptrons. Στην εικόνα 3.4 στο σχήμα *B* παρουσιάζεται η δομή ενός perceptron, ενώ στο σχήμα *D* φαίνεται ένα τυπικό νευρωνικό δίκτυο αποτελούμενο από πολλά στοιβαγμένα perceptrons.

Τρία διαφορετικά (στρώματα-)επίπεδα απαρτίζουν ένα Νευρωνικό Δίκτυο, όπως παρουσιάζεται στην εικόνα 3.5:

1. Input Layer: Οι εισόδους τροφοδοτούνται στο μοντέλο του νευρωνικού δικτύου μέσω αυτού του επιπέδου.
2. Hidden Layer: Οι κόμβοι του συγκεκριμένου επιπέδου επεξεργάζονται τις εισόδους του προηγούμενου επιπέδου. Το νευρωνικό δίκτυο μπορεί να περιλαμβάνει ένα ή περισσότερα επίπεδα από κρυφούς κόμβους.
3. Output Layer: Το δίκτυο τροφοδοτεί το υπόλοιπο σύστημα με τα αποτελέσματά του μέσω αυτού του επιπέδου.

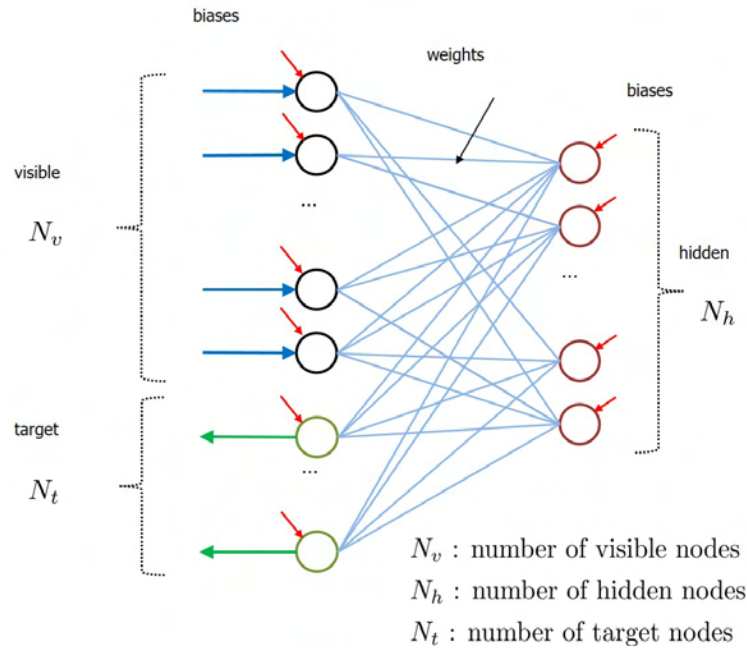
3.3 Classification Restricted Boltzmann Machines



Σχήμα 3.5: Γενική Αρχιτεκτονική Νευρωνικού Δικτύου και *stacked RBM*

Μια μηχανή *Boltzmann* είναι ένα δίκτυο συμμετρικά συνδεδεμένων μονάδων που έχουν λειτουργία ανάλογη με αυτή των perceptrons. Συγκεκριμένα, κάθε μονάδα παίρνει στοχαστικές αποφάσεις σχετικά με το αν πρέπει να είναι ενεργοποιημένη ή απενεργοποιημένη. Τα Boltzmann Machines ανήκουν στην κατηγορία των πιθανοτικών μοντέλων Markov Random Fields. Οι *visible nodes* συνιστούν το πρώτο επίπεδο του δικτύου και αντιστοιχούν στα συστατικά μιας παρατήρησης (π.χ. μία ορατή μονάδα για κάθε εικονοστοιχείο (pixel) μιας ψηφιακής εικόνας εισόδου). Οι *hidden nodes* μοντελοποιούν τις εξαρτήσεις μεταξύ των συνιστωσών των παρατηρήσεων, ή αλλιώς των *visible nodes*. Παραδείγματος χάρι οι εξαρτήσεις

ανάμεσα στα εικονοστοιχεία μιας εικόνας. Οι hidden nodes μπορούν να θεωρηθούν ως ανιχνευτές χαρακτηριστικών (feature detectors). Τέλος, οι target nodes αποτελούν τις εξόδους του δικτύου.



Σχήμα 3.6: Αρχιτεκτονική δικτύου RBM

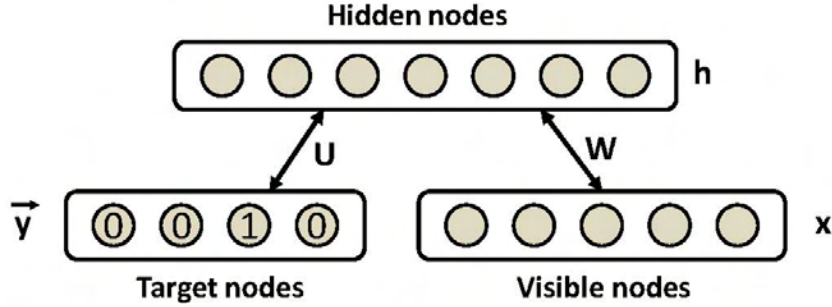
Τα *Restricted Boltzmann Machines (RBMs)* αποτελούν μία ειδική κατηγορία των Boltzmann Machines [8], [9]. Κάθε κύκλος στο γράφημα 3.6 αντιπροσωπεύει μια μονάδα που μοιάζει με ένα perceptron και ονομάζεται *κόμβος (node)*. Σε κάθε κόμβο πραγματοποιούνται συγκεκριμένοι υπολογισμοί. Ουσιαστικά στα RBMs δίκτυα εισάγεται ο περιορισμός ότι *δεν επιτρέπεται η σύνδεση κόμβων του ίδιου είδους μεταξύ τους*. Ο λόγος για την επιβολή του συγκεκριμένου περιορισμού στην τοπολογία του δικτύου, είναι η μείωση του όγκου των απαιτούμενων υπολογισμών κατά το στάδιο της *εκπαίδευσης* του δικτύου.

Η δομή ενός RBM νευρωνικού δικτύου παρουσιάζεται στο γράφο του σχήματος 3.6. Στο σχήμα αυτό εκτός από τους visible, hidden και target nodes φαίνονται τα *Weights* και *Biases* του δικτύου. Η φυσική σημασία των *Weights* εκφράζει τη βαρύτητα μίας σύνδεσης-συναψης ανάμεσα σε δύο κόμβους. Επιπρόσθετα τα *Weights* προσφέρουν μία εικόνα των εξαρτήσεων του δικτύου και παίζουν ρόλο στην στοχαστική απόφαση που λαμβάνεται από κάθε κόμβο για την κατάστασή του, αν δηλαδή ενεργοποιείται ή απενεργοποιείται. Η δυαδική κατάσταση στην οποία μπορεί να βρεθεί κάθε τεχνητός νευρώνας σε ένα δίκτυο RBM ορίζει και τη φύση αυτών των στοιχείων. Συγκεκριμένα σε ένα RBM νευρωνικό δίκτυο τόσο οι κόμβοι εισόδου (visible nodes) και εξόδου (target nodes), όσο και οι hidden nodes μπορούν να πάρουν τιμές στο δυαδικό σύνολο 0, 1, αποτελούν δηλαδή δυαδικά στοιχεία. Σε επόμενο κεφάλαιο του παρόντος συγγράμματος θα εξηγηθεί πώς ο περιορισμός επηρεάζει τις αποφάσεις που λαμβάνονται από τη σκοπιά της σχεδίασης για υλοποίηση ενός δικτύου RBM.

Οι Restricted Boltzmann Machines συχνά χρησιμοποιούνταν ως feature extractors για άλλους learning αλγορίθμους ή ως δίκτυα αρχικοποίησης για *deep feedforward neural network classifiers*. Όχι όμως ως αυτόνομοι classifiers. Παρόλα αυτά οι συγγραφείς του [10] παρου-

σιάζουν μία διαφορετική προσέγγιση των δικτύων RBMs που μπορεί να ανταποκριθεί στο πρόβλημα του classification προσφέροντας αποδεκτή ακρίβεια.

3.3.1 RBM Training και Classification



Σχήμα 3.7: Μοντελοποίηση της από κοινού κατανομής των Visible και Target nodes με χρήση Discriminative RBM

Η εκμάθηση ενός RBM δικτύου αντιστοιχεί στην προσαρμογή των παραμέτρων του έτσι ώστε η κατανομή που αντιπροσωπεύεται από το δίκτυο να προσεγγίζει την κατανομή που εμφανίζουν τα δεδομένα εκπαίδευσης. Δίνεται ένα training set $D_{train} = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_D, y_D)\}$, όπου το i είναι το i -οστό παράδειγμα από ένα σύνολο που αποτελείται από διανύσματα εισόδου x_i και αναμενόμενες εξόδους $y_i \in \{1, \dots, C\}$. Χρησιμοποιείται ένα δίκτυο RBM με στόχο τη μοντελοποίηση της από κοινού κατανομής των N τον αριθμό *hidden* μεταβλητών $\mathbf{h} = (h_1, \dots, h_N)$ και των μεταβλητών εισόδου και εξόδου $(\mathbf{x}, \mathbf{y})$. Το μοντέλο που αναπτύσσεται είναι ένα παραμετρικό μοντέλο όπου οι παράμετροι $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{U})$ αντιπροσωπεύουν τα ακόλουθα:

- **W:** Πίνακας των weights ανάμεσα στις μεταβλητές $\mathbf{x}$ και $\mathbf{h}$
- **U:** Πίνακας των weights ανάμεσα στις μεταβλητές e_y και $\mathbf{h}$
- **b, c, d:** Τα αντίστοιχα biases των μεταβλητών $\mathbf{x}$, $\mathbf{h}$ και e_y

και η μεταβλητή $e_y = (1_{i=y})$ για $i \in \{1, \dots, C\}$ αποτελεί τη διανυσματική αναπαράσταση της εξόδου y .

Θεωρούμε τη δυαδική έκδοση του μοντέλου, όπως αναφέραμε προηγουμένως, όπου κάθε visible, hidden ή target node, μπορεί να είναι **ON** ή **OFF**. Κάθε κόμβος υιοθετεί μία νέα κατάσταση ως πιθανοτική συνάρτηση της κατάστασης των γειτονικών κόμβων και των weights που είναι συνδεδεμένα μαζί του. Οι παρακάτω εξισώσεις περιγράφουν την εν λόγω πιθανότητα, για κάθε κόμβο:

$$p(h_j = 1 | y, x) = \text{sigm}(c_j + u_{jy} + \sum_i w_{ji} x_i) \quad (3.1)$$

$$p(x_i = 1 | h) = \text{sigm}(b_i + \sum_j w_{ji} h_j) \quad (3.2)$$

$$p(y|h) = \frac{\exp(d_y + \sum_j u_{jy} h_j)}{\sum_{\text{for all } y} \exp(d_y + \sum_j u_{jy} h_j)} \quad (3.3)$$

όπου $\text{sigm}(x) = 1/(1 + e^{-x})$ είναι η *logistic sigmoid function*.

Μετά τη διαδικασία του training του μοντέλου, η υπο συνθήκη πιθανότητα $p(y|x)$ χρησιμοποιείται για τη διαδικασία του classification. Η συγκεκριμένη πιθανότητα $p(y|x)$ μπορεί να υπολογιστεί με τη χρήση της εξίσωσης $p(y|x) = \text{argmin}(F(y, x))$:

$$F(y, h) = -d_y - \sum_j \log(1 + e^{(c_j + u_{jy} + \sum_i w_{ji} x_i)}) \quad (3.4)$$

όπου η συνάρτηση $F(y, h)$ ονομάζεται *free energy*. Η φυσική σημασία της διαδικασίας του classification με τη χρήση της free energy function, αντιστοιχεί στην εύρεση και επιλογή του target node ο οποίος εμφανίζει την ελάχιστη ενέργεια για το δίκτυο ([11],[12]).

Για την εκπαίδευση ενός δικτύου RBM με κατεύθυνση τη λύση ενός ειδικού προβλήματος classification, πρέπει να ορισθεί ως στόχος της διαδικασίας εκμάθησης η ελαχιστοποίηση μίας συγκεκριμένης συνάρτησης για όλα τα παραδείγματα που συμπεριλαμβάνονται στο D_{train} [13]. Είναι δυνατό να επιλεχθούν διάφορα μεγέθη ως συνάρτησεις προς ελαχιστοποίηση. Παρακάτω παρουσιάζονται 3 αλγόριθμοι που σχετίζονται με αυτήν την επιλογή:

- Generative Training Objective
- Discriminative Training Objective
- Hybrid Training Objective

3.3.2 Generative Training Objective

Με δεδομένο ότι το μοντέλο καθορίζει μία από κοινού πιθανότητα $p(x, y)$, μία φυσική επιλογή ως κριτήριο για τη διαδικασία της εκπαίδευσης αποτελεί το *generative objective*, η ελαχιστοποίηση δηλαδή της παρακάτω συνάρτησης:

$$L_{gen}(D_{train}) = - \sum_{i=1}^{|D_{train}|} \log p(y_i, x_i) \quad (3.5)$$

Ο υπολογισμός του $\log p(y_i, x_i)$ και των παραγώγων του (*gradient*), είναι απαγορευτική από τη σκοπιά των απαιτούμενων υπολογισμών. Ωστόσο έχει αποδειχθεί ότι η παράγωγος (*gradient*) της $\log p(y_i, x_i)$ μπορεί να προσεγγιστεί με καλή ακρίβεια με τη χρήση του *Contrastive Divergence estimator* [14]. Ο αναλυτικός υπολογισμός αντικαθίσταται από μία προσέγγιση, η οποία προκύπτει από περιορισμένο αριθμό βημάτων δειγματοληψίας με τη χρήση του αλγορίθμου *Gibbs sampling* [9]. Για την αρχικοποίηση των visible και target nodes αξιοποιούνται ως είσοδοι τα παραδείγματα (x_i, y_i) του D_{train} . Μία επανάληψη του συγκεκριμένου αλγορίθμου είναι συνήθως ικανή να μοντελοποιήσει τα δεδομένα εισόδου. Στη συνέχεια αυτή η προσέγγιση μπορεί να χρησιμοποιηθεί για τη διαδικασία gradient descent στο στάδιο του training. Ο αλγόριθμος σε μορφή ψευδοκώδικα που ακολουθεί, περιγράφει τη συγκεκριμένη διαδικασία:

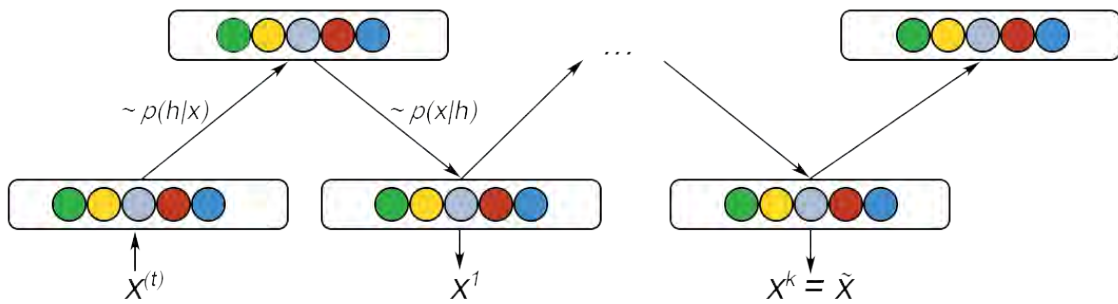
 ΑΛΓΟΡΙΘΜΟΣ 3.1: *RBM Training over (x,y) using 1-step Contrastive Divergence*

Input: w_{ij}
for all training samples (x, y) **do**

1. Calculate $p(h = 1|x, y)$ for all hidden nodes
2. Sample the hidden distribution $\langle h_0 \rangle$
3. Perform Gibbs sampling for k steps ($k=1$):
 Calculate $p(x|h_0)$ for all visible nodes and $p(y|h_0)$ for all target nodes
 Sample the visible distribution $\langle x_k \rangle$ and the target distribution $\langle y_k \rangle$
 Calculate $p(h = 1|x_k, y_k)$ for all hidden nodes
 Sample the hidden distribution $\langle h_k \rangle$
4. Calculate gradients:
 $gW = \langle h_k \rangle * \langle x_k \rangle - \langle h_0 \rangle * x$
 $gU = \langle h_k \rangle * \langle y_k \rangle - \langle h_0 \rangle * y$
 $gb = \langle x_k \rangle - x$
 $gc = \langle y_k \rangle - y$
 $gd = \langle h_k \rangle - \langle h_0 \rangle$
5. Update weights and biases:
 $W' = W - \gamma * gW$
 $U' = U - \gamma * gU$
 $b' = b - \gamma * gb$
 $c' = c - \gamma * gc$
 $d' = d - \gamma * gd$

end for

Ως γ ορίζεται η παράμετρος *learning rate*. Συνήθως τα weights (**W,U**) αρχικοποιούνται με μικρές τυχαίες τιμές, ενώ τα biases (**b,c,d**) παίρνουν μηδενικές τιμές. Ιδανικά, τα δίκτυα RBMs απαιτούν ανανέωση των παραμέτρων τους για *κάθε* παράδειγμα εισόδου. Παρόλα αυτά, η ανανέωση των παραμέτρων μπορεί να εκτελεστεί και ανά δέσμη (batch) από παραδείγματα. Με σκοπό να εξασφαλιστεί ταχεία σύγκλιση (convergence) για το μοντέλο, ο αριθμός των παραδειγμάτων ανα δέσμη πρέπει να διατηρηθεί μικρός.



Σχήμα 3.8: Αλγόριθμος $k - step$ Contrastive Divergence

3.3.3 Discriminative Training Objective

Ο αλγόριθμος generative training objective μπορεί να τροποποιηθεί ως εξής:

$$L_{gen}(D_{train}) = - \sum_{i=1}^{|D_{train}|} \log p(y_i|x_i) - \sum_{i=1}^{|D_{train}|} \log p(x_i) \quad (3.6)$$

Αυτό σημαίνει ότι ο RBM classifier αφιερώνει ένα κομμάτι των υπολογισμών του στον υπολογισμό μόνο της marginal distribution των δειγμάτων εισόδου. Με αυτόν τον τρόπο η συνάρτηση προς ελαχιστοποίηση ορίζεται ως εξής:

$$L_{disc}(D_{train}) = - \sum_{i=1}^{|D_{train}|} \log p(y_i|x_i) \quad (3.7)$$

Το σημαντικότερο πλεονέκτημα αυτής της μεθόδου εκπαίδευσης αποτελεί το γεγονός ότι καθίσταται δυνατός ο επακριβής υπολογισμός των δειγμάτων της παραγώγου (gradient) για κάθε (x_i, y_i) παράδειγμα εισόδου. Για τις παραμέτρους του μοντέλου ισχύουν οι παρακάτω εξισώσεις:

- Για $\Theta = (\mathbf{c}, \mathbf{W}, \mathbf{U})$:

$$\begin{aligned} \frac{\partial \log p(y_i|x_i)}{\partial \Theta} &= \sum_j \text{sigm}(O_{y_i,j}(x_i)) \frac{\partial O_{y_i,j}(x_i)}{\partial \Theta} \\ &\quad - \sum_{j, \text{forall}} \text{sigm}(O_{y,j}(x_i)) p(y|x_i) \frac{\partial O_{y,j}(x_i)}{\partial \Theta} \end{aligned} \quad (3.8)$$

όπου $O_{y,j}(x) = c_j + u_{jy} + \sum_i w_{ji}x_i$ για τον hidden node j .

- For $\Theta = (\mathbf{d})$:

$$\frac{\partial \log p(y_i|x_i)}{\partial d_y} = 1_{y=y_i} - p(y|x_i), \forall y \in \{1, \dots, C\} \quad (3.9)$$

- Για $\Theta = (\mathbf{b})$ η παράγωγος είναι μηδέν, από τη στιγμή που τα biases δε συμμετέχουν στον υπολογισμό της πιθανότητας $p(y|x)$.

3.3.4 Hybrid Training Objective

Η αποτελεσματικότητα των δύο προηγούμενων μεθόδων έχει μελετηθεί και αποδείχθηκε ότι εμφανίζουν πολύ διαφορετικά χαρακτηριστικά. Για τη διαδικασία του classification, η προσθήκη ενός όρου της συνάρτησης του generative training objective είναι ικανή να εξομαλύνει τις ανισοροπίες του discriminative training objective αλγορίθμου. Με αυτόν τον τρόπο προκύπτει η μέθοδος του Hybrid Training Objective η οποία προσπαθεί να ελαχιστοποιήσει τη συνάρτηση:

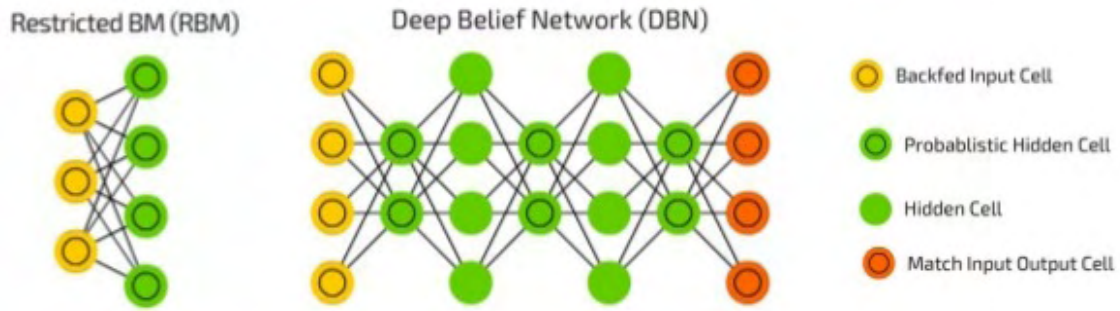
$$L_{hybrid}(D_{train}) = L_{disc}(D_{train}) - \alpha L_{gen}(D_{train}) \quad (3.10)$$

όπου το βάρος α του generative μέρους μπορεί να προσαρμοστεί ανάλογα με την απόδοση του μοντέλου με χρήση ενός συγκεκριμένου validation set.

Η συγκεκριμένη διπλωματική εργασία ασχολήθηκε με την υλοποίηση της διαδικασίας του classification σε αντίθεση με παρόμοιες εργασίες που εστίασαν στο πρόβλημα της επιτάχυνσης

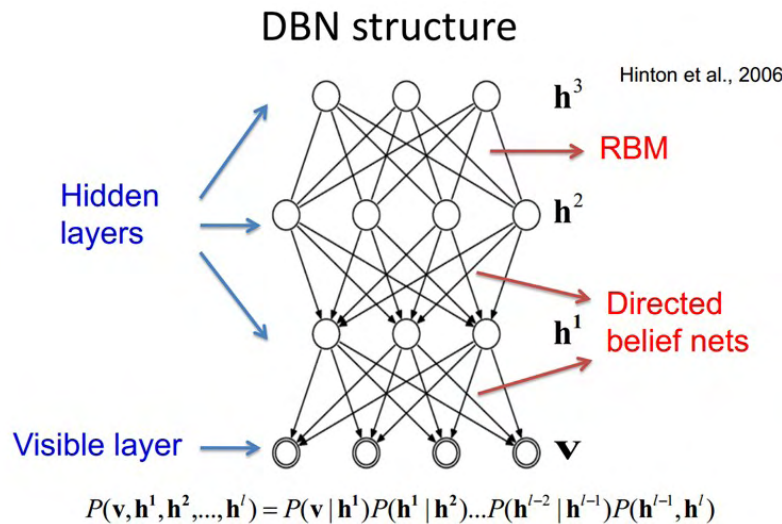
της διαδικασίας του training ([15]). Ωστόσο οι απαραίτητοι παράμετροι για το σύστημα, οι οποίοι προκύπτουν από τη διαδικασία της εκπαίδευσης, παρήχθησαν με τη χρήση του Hybrid Training Objective αλγορίθμου [18].

3.4 Deep Learning - Deep Belief Networks



Σχήμα 3.9: Δομή δικτύου DBN

Όπως αναλύθηκε για την περίπτωση των RBM δικτύων, οι τεχνητοί νευρώνες συνδέονται με συνάψεις, η ισχύς των οποίων περιγράφεται μέσω βαρών. Για ένα δίκτυο με πεπερασμένο αριθμό νευρώνων, κάθε ένας εκτελεί κάποιου είδους υπολογισμό και στη συνέχεια το αποτέλεσμα πολλαπλασιάζεται με έναν συντελεστή. Ο συντελεστής αυτός καθορίζει τη βαρύτητα της διασύνδεσης. Το σταθμισμένο αποτέλεσμα που προκύπτει συνήθως αποτελεί την έξοδο του νευρωνικού δικτύου. Στην περίπτωση δικτύων τα οποία συντάσσονται κλιμακωτά, με τρόπο τέτοιο ώστε η έξοδος του ενός να χρησιμοποιείται σαν είσοδο στο επόμενο, αναπτύσσονται δομές που είναι γνωστές ως *Deep belief Networks* (DBN) [16], .



Σχήμα 3.10: Δίκτυο DBN 3 Hidden Layers

Η αρχιτεκτονική ενός τέτοιου δικτύου περιγράφεται από τις εικόνες 3.9 και 3.10, όπου παρουσιάζεται η τοπολογία με την οποία συντάσσονται τα διαδοχικά επίπεδα των δικτύων RBM για τη διαμόρφωση ενός DBN δικτύου. Τα δίκτυα *Deep belief Networks* έχουν τρα-

βήξει το ενδιαφέρον της επιστημονικής κοινότητας τα τελευταία χρόνια, κυρίως λόγω των δυνατοτήτων που φαίνεται να προσφέρουν στην αναγνώριση προτύπων. Ωστόσο, η χρήση τέτοιων αλγορίθμων για τη λύση ενός πραγματικού προβλήματος δεν είναι καθόλου τετριμμένη υπόθεση. Οι απαιτήσεις υλοποίησης τέτοιων αλγορίθμων σε αποθήκευση και υπολογιστική ισχύ είναι τεράστιες. Για το λόγο αυτό είναι κομβικής σημασίας η συνεχής βελτιστοποίηση των δικτύων σε αλγοριθμικό επίπεδο αλλά και η επέκταση της ανάπτυξης συστημάτων μηχανικής μάθησης με χρήση σύγχρονης τεχνολογίας. Η επιστράτευση ισχυρών επεξεργαστών, πολυεπεξεργαστικών συστημάτων, *GPUs* αλλά και μονάδων εξειδικευμένου υλικού (*ASIC*, *FPGAs*) είναι επιβεβλημένη ([17]).

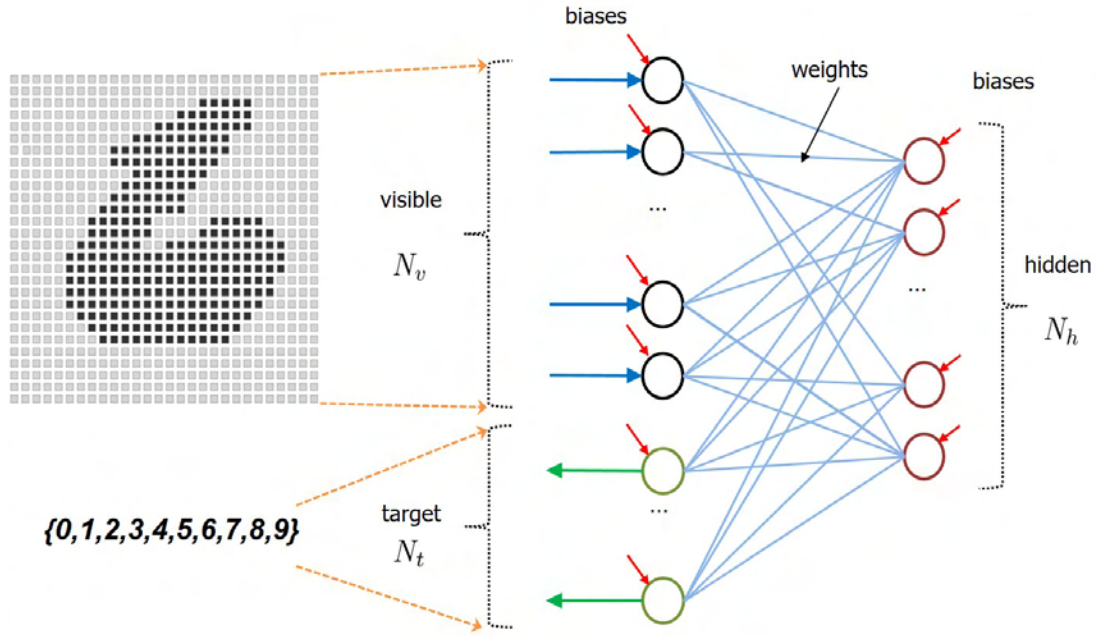
Κεφάλαιο 4

RBM στην Αναγνώριση Χειρόγραφων Ψηφίων

Τα δύο προηγούμενα κεφάλαια ασχολήθηκαν με τις διαθέσιμες βιβλιοθήκες χειρόγραφων ψηφίων αλλά και την παρουσίαση των αλγορίθμων Μηχανικής Μάθησης και συγκεκριμένα των δικτύων RBM. Στο κεφάλαιο αυτό θα περιγραφούν οι τρόποι που μπορούν να αξιοποιηθούν τα RBM νευρωνικά δίκτυα για την επίλυση του προβλήματος της αναγνώρισης χειρόγραφων ψηφίων. Ακόμα θα παρουσιαστεί η ανάλυση που έγινε στο πλαίσιο της διπλωματικής εργασίας, με στόχο τον καθορισμό διαφόρων παραμέτρων του συστήματος, για τη βέλτιστη σχεδίαση και υλοποίηση του *Handwritten Digits Recognition-RBM (HDR-RBM) Hardware Accelerator*.

4.1 Μοντέλο Προσομοίωσης HDR-RBM

Κατά το στάδιο της σχεδίασης, ένα από τα σημαντικότερα βήματα αποτελεί η μοντελοποίηση του συστήματος προς υλοποίηση. Το μοντέλο δίνει τη δυνατότητα για επαλήθευση της ορθής λειτουργίας του αλγορίθμου και επιτρέπει την ανάλυση, για το βέλτιστο καθορισμό των παραμέτρων. Σε ένα σύστημα προσομοίωσης μπορούν να εφαρμοστούν αλλαγές και να μετρηθεί η επιρροή τους στην απόδοση και την ακρίβεια των υπό εξέταση αλγορίθμων. Στη συγκεκριμένη διπλωματική εργασία αναπτύχθηκε, με τη χρήση του εργαλείου *MATLAB*, ένα μοντέλο προσομοίωσης ενός RBM νευρωνικού δικτύου, το οποίο δέχεται σαν είσοδο αρχεία που περιέχουν εικόνες από τη βιβλιοθήκη του MNIST. Το μοντέλο επεξεργάζεται τις εικόνες και εκτελώντας τη διαδικασία του Classification, επιστρέφει ως αποτέλεσμα τα αναγνωρισμένα ψηφία. Ακόμα προσφέρει στοιχεία που σχετίζονται με την ακρίβεια του αλγορίθμου καθώς και με τον χρόνο εκτέλεσης σε επίπεδο προσομοίωσης. Το μοντέλο αυτό αξιοποιήθηκε για την ανάλυση διαφόρων μεταβλητών του συστήματος. Έγιναν μία σειρά δοκιμές για τις τελικές αποφάσεις και τον καθορισμό των παραμέτρων της αρχιτεκτονικής του hardware accelerator, όπως για παράδειγμα ο βαθμός παραλληλοποίησης του αλγορίθμου. Περισσότερα θα παρουσιαστούν στις ενότητες που ακολουθούν.



Σχήμα 4.1: Εφαρμογή HDR σε δίκτυο RBM

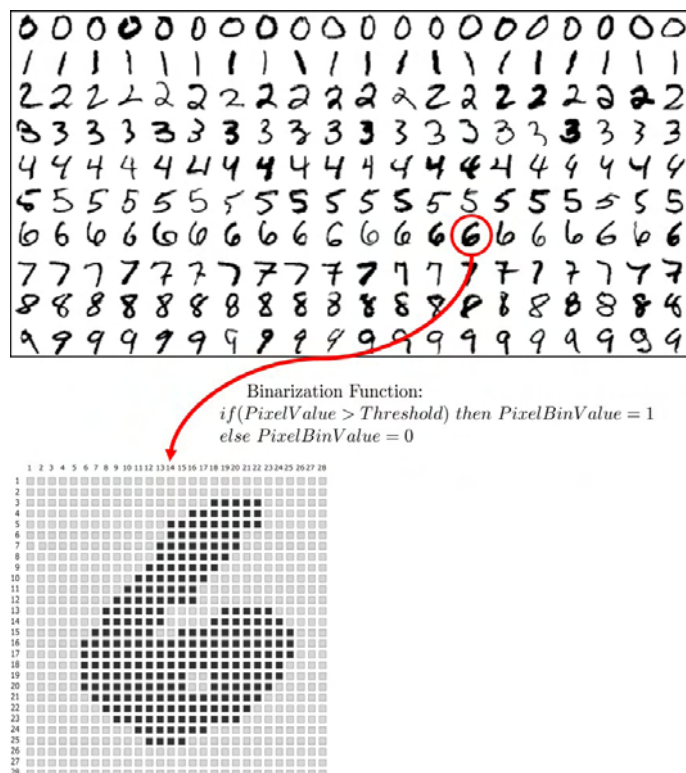
4.2 Ανάλυση παραμέτρων συστήματος

4.2.1 Πλήθος N_t κόμβων

Η εικόνα 4.1 δίνει μία αίσθηση της εφαρμογής του αλγορίθμου των δικτύων RBM στην αυτόματη αναγνώριση χειρόγραφων ψηφίων, σε επίπεδο εισόδων και εξόδων. Με δεδομένο ότι ο αριθμός των ψηφίων προς αναγνώριση είναι 10, καθώς αναγνωρίζονται ψηφία από το 0 μέχρι το 9, η μεταβλητή N_t παίρνει την τιμή 10. Άρα αυτό είναι και το πλήθος των target nodes, που στην περίπτωση του HDR-RBM αξιοποιούνται ως έξοδοι. Αναφορικά με τις παραμέτρους Weights και Biases, οι οποίες υπολογίζονται κατά το στάδιο του training ενός νευρωνικού δικτύου, αυτές προσφέρονται αποθηκευμένες σε μορφή πινάκων και αξιοποιούνται στα αντίστοιχα σκέλη του αλγορίθμου του classification. Οι τιμές τους ανήκουν στο σύνολο των πραγματικών αριθμών και άρα μπορούν να είναι είτε θετικές είτε αρνητικές.

4.2.2 Μέγεθος εικόνων - Πλήθος N_v κόμβων

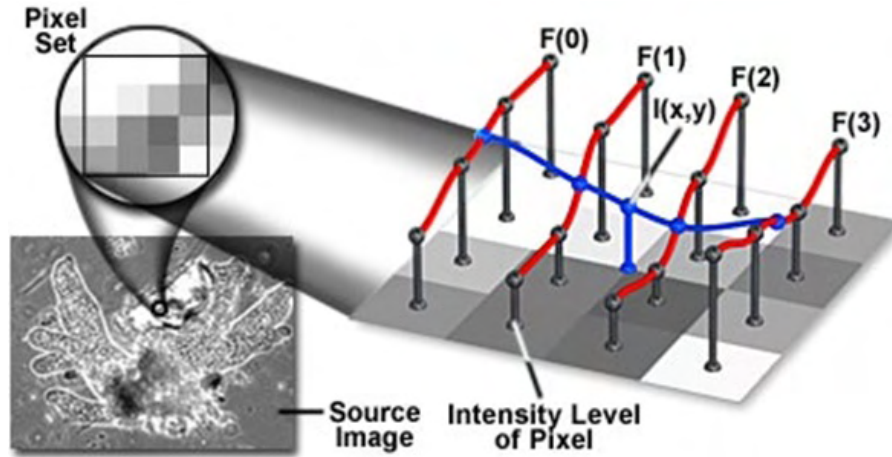
Μία ιδιαίτερης σημασίας παράμετρος που πρέπει να επιλεγθεί για το συνολικό σύστημα, είναι αυτή του μεγέθους των εικόνων που χρησιμοποιούνται ως είσοδοι. Όπως περιγράφηκε στο κεφάλαιο 2, χρησιμοποιήθηκε το MNIST Dataset. Το μέγεθος των εικόνων του MNIST είναι 28×28 pixels. Κάθε pixel αντιπροσωπεύεται από έναν αριθμό των 8 bits, ο οποίος παίρνει τιμές από 0–255 σε σύστημα greyscale. Για την υλοποίηση του HDR-RBM δικτύου, οι visible nodes ταυτίζονται με τα pixels της εκάστοτε εικόνας προς αναγνώριση. Όπως περιγράφηκε στο κεφάλαιο 2, από τον ορισμό των RBM νευρωνικών δικτύων, οι visible nodes είναι δυαδικές μεταβλητές, παίρνουν δηλαδή τιμές 0 ή 1. Άρα, λοιπόν, για λόγους συμβατότητας, πρέπει να εκτελεστεί μία μετατροπή, ώστε τα pixels, από 8 bits αριθμοί να γίνουν και αυτοί δυαδικοί. Η



Σχήμα 4.2: MNIST Binarization technique

δυναδικασία αυτή ονομάζεται *Binarization* και συμβαίνει στο στάδιο της προσομοίωσης. Στο σχήμα φαίνεται οπτικά ο τρόπος που εφαρμόζεται το binarization. Ουσιαστικά η τιμή κάθε pixel συγκρίνεται με ένα συγκεκριμένο κατώφλι και με βάση αυτή τη σύγκριση προκύπτει η τιμή 0 ή 1. Με αυτό τον τρόπο ανάπτυσσονται εικόνες που έχουν pixels με δυαδικούς αριθμούς. Τα παραπάνω φαίνονται στην εικόνα 4.2. Αξίζει να σημειωθεί ότι για την επιλογή της τιμής του κατωφλίου, διεξήχθη ανάλυση η οποία βασίστηκε στο κριτήριο της ακρίβειας με στόχο τη βέλτιστη τιμή. Συγκεκριμένα, με δεδομένες τις εικόνες του *Test MNIST Dataset*, μετρήθηκε ο αριθμός των λαθών που προέκυπτε από τη χρήση κάθε τιμής κατωφλίου στο σύνολο 0,255 και με βήμα 1. Βρέθηκε ότι η τιμή **128** δίνει το μικρότερο αριθμό λαθών και άρα τη μέγιστη ακρίβεια.

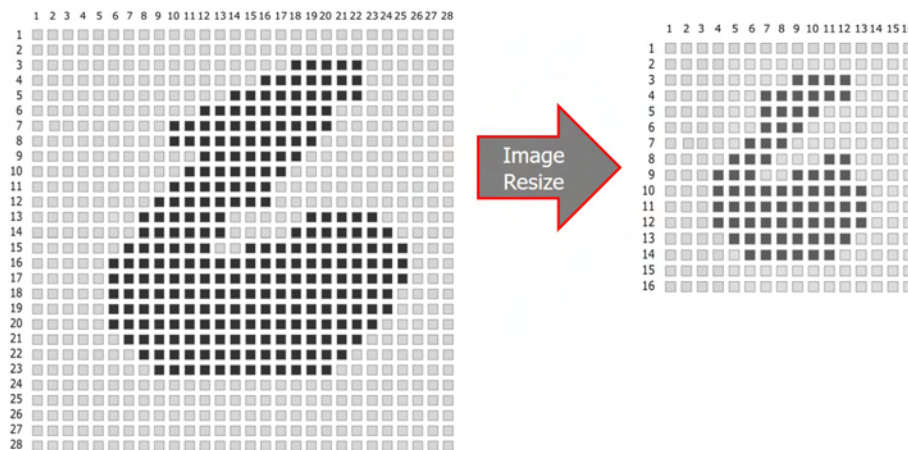
Με βάση τις εξισώσεις του Classification, που παρουσιάστηκαν στο κεφάλαιο 3, η πολυπλοκότητα των υπολογισμών αυξάνεται ανάλογα με τον αριθμό των visible nodes και άρα είναι τάξης $O(n)$, όπου $n = N_v$. Άρα, μία μείωση σε αυτόν τον αριθμό μπορεί να οδηγήσει, σε μείωση των απαιτούμενων πράξεων. Η μείωση αυτή μπορεί να επιφέρει περαιτέρω ελάττωση τόσο στον απαιτούμενο χρόνο εκτέλεσης, όσο και στους πόρους του συστήματος, από τη σκόπια του υλικού. Τα στοιχεία αυτά έχουν πρωτεύουσα σημασία στην κατασκευή ενός hardware κυκλώματος, πόσο μάλλον όταν αυτό προορίζεται για την επιτάχυνση ενός αλγορίθμου, όπου στόχος αποτελεί το μέγιστο δυνατό throughput. Για τον σκοπό αυτό, μελετήθηκε η επίδραση στην ακρίβεια του classification, μετά την εφαρμογή διαφόρων μηχανισμών μείωσης του μεγέθους των εικόνων. Βασική μέθοδος που αξιοποιήθηκε είναι αυτή της *Παραμβολής (Interpolation)*. Συγκεκριμένα, ύστερα από σύγκριση των αποτελεσμάτων που προέκυπταν από κάθε μηχανισμό, βρέθηκε ότι τα βέλτιστα αποτελέσματα, για το *Image Size Reduction*,



Σχήμα 4.3: Bicubic Interpolation

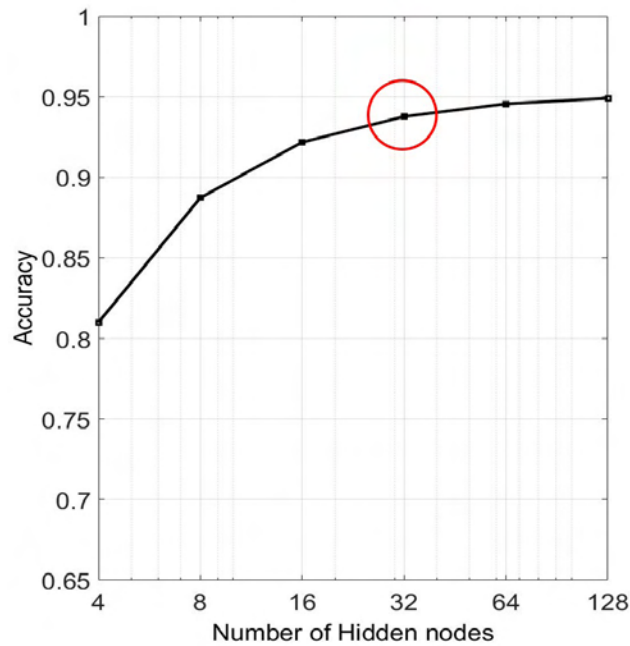
επιτυγχάνονται από τη μέθοδο του *Bicubic Interpolation*, η οποία περιγράφεται από το σχήμα 4.3. Για τη σύγκριση χρησιμοποιήθηκε ως κριτήριο η ακρίβεια του classification με είσοδο το σύνολο των εικόνων που προέκυπταν μετά το *Interpolation*. Η τελική επιλογή των $N_v = 256$ visible nodes ήταν αποτέλεσμα σύμβασης ανάμεσα στην ακρίβεια του συστήματος και την πολυπλοκότητα. Παράδειγμα μείωσης του μεγέθους της εικόνας φαίνεται στο σχήμα 4.4. Με βάση την παραπάνω ανάλυση, αναπτύχθηκαν με τη χρήση του εργαλείου MATLAB, νέα *reduced MNIST Datasets*, τα οποία έχουν τον ίδιο αριθμό εικόνων, μεγέθους 16×16 pixels.

Ταυτόχρονα με τη μείωση της πολυπλοκότητας και του απαιτούμενου latency για τους υπολογισμούς, η μείωση του μεγέθους των εικόνων εισόδου παρουσιάζει το πλεονέκτημα του μικρότερου απαιτούμενου Data Rate για τις απαραίτητες μεταφορές δεδομένων. Στοιχεία που αφορούν τις μεταφορές αυτές, καθώς και οι σχετικές μετρήσεις απόδοσης του συστήματος θα παρουσιαστούν στο κεφάλαιο 7.



Σχήμα 4.4: Μείωση μεγέθους εικόνων: από 784 σε 256 pixels

4.2.3 Πλήθος N_h κόμβων



Σχήμα 4.5: Ακρίβεια RBM vs N_h

Με δεδομένο το πλήθος των visible και target nodes, μένει να καθοριστεί το πλήθος των hidden nodes. Αντίστοιχα με τους visible nodes, οι εξισώσεις του αλγορίθμου παρουσιάζουν πολυπλοκότητα ανάλογη με το πλήθος των hidden nodes, τάξης $O(n)$, όπου $n = N_h$. Έτσι, λοιπόν, όπως και στην περίπτωση των visible nodes, παίζει μεγάλο ρόλο στην απόδοση του τελικού συστήματος η βέλτιστη επιλογή του N_h αριθμού. Πρέπει να τονιστεί ότι η ακρίβεια ενός RBM classification αλγορίθμου, αποτελεί μία συνάρτηση του συνδυασμού του πλήθους των visible και hidden nodes. Για παράδειγμα, στην περίπτωση που ως εικόνες εισόδου επιλέγονταν αυτές του original MNIST Dataset, με μέγεθος 784 pixels, πειράματα έδειξαν ότι για ικανοποιητική ακρίβεια, ο αριθμός των hidden nodes έπρεπε να παίρνει τιμές μεγαλύτερες από 90. Ωστόσο για αντίστοιχη ακρίβεια, με κάποια μικρή μείωση φυσικά, και με τη χρήση των *reduced MNIST Datasets*, ο αριθμός αυτός έπεφτε ακόμα και στο $N_h = 16$. Στο διάγραμμα 4.5 φαίνεται η σχέση της ακρίβειας του αλγορίθμου με το πλήθος των hidden nodes. Από το σχήμα μπορούν να εξαχθούν τα εξής συμπεράσματα:

1. Η ακρίβεια του συστήματος αυξάνεται με τον αριθμό των hidden nodes μέχρι μία μέγιστη τιμή.
2. Από τη μέγιστη τιμή και έπειτα, δεν έχει νόημα η περαιτέρω αύξηση του N_h .
3. Το σύστημα με τη χρήση του *reduced MNIST Datasets* προσφέρει μέγιστη ακρίβεια 95.1%. Η αντίστοιχη μέγιστη ακρίβεια για το original MNIST Dataset είναι 97% για $N_h = 200$.

Ο πίνακας 4.1 παρουσιάζει την απόκριση του αλγορίθμου αναγνώρισης ψηφίων σε διαφορετικούς συνδυασμούς των παραμέτρων N_v και N_h . Συγκεκριμένα φαίνεται η ακρίβεια για

<i>Dataset – N_h</i>	N_v	N_h	<i>Accuracy</i>	<i>Complexity ($N_v \times N_h$)</i>
MNIST - 98	784	98	0.959	76,832
<i>Modified – 4</i>	256	4	0.810	1,024
<i>Modified – 8</i>	256	8	0.887	2,048
<i>Modified – 16</i>	256	16	0.922	4,096
Modified - 32	256	32	0.938	8,192
<i>Modified – 64</i>	256	64	0.946	16,384
<i>Modified – 128</i>	256	128	0.951	32,768

Πίνακας 4.1: Ακρίβεια και Πολυπλοκότητα σε συνάρτηση με N_v και N_h

κάθε διαφορετικό συνδυασμό Dataset- N_h , όπου το αντίστοιχο Dataset αξιοποιείται ως είσοδος. Στον ίδιο πίνακα φαίνεται η απαιτούμενη πολυπλοκότητα για τις διαφορετικές περιπτώσεις συνδυασμών Dataset- N_h . Από τον πίνακα, όπως και από τη γραφική παράσταση 4.5, γίνεται κατανοητό ότι η τελική επιλογή των $N_h = 32$ hidden nodes είναι η βέλτιστη σύμβαση ανάμεσα στην ακρίβεια που επιτυγχάνεται και την πολυπλοκότητα. Αξίζει να σημειωθεί ότι, παρατηρώντας τον πίνακα 4.1 και συγκρίνοντας τις περιπτώσεις του *original MNIST Dataset*, **MNIST - 98** και του *reduced MNIST Dataset, Modified - 32*, παρουσιάζεται μικρή μείωση στην ακρίβεια, της τάξης του 2.2%, ενώ ταυτόχρονα εμφανίζεται μείωση στην πολυπλοκότητα, της τάξης του 900%.

4.3 Επίδραση της τροποποίησης των MNIST Datasets στην ακρίβεια του συστήματος

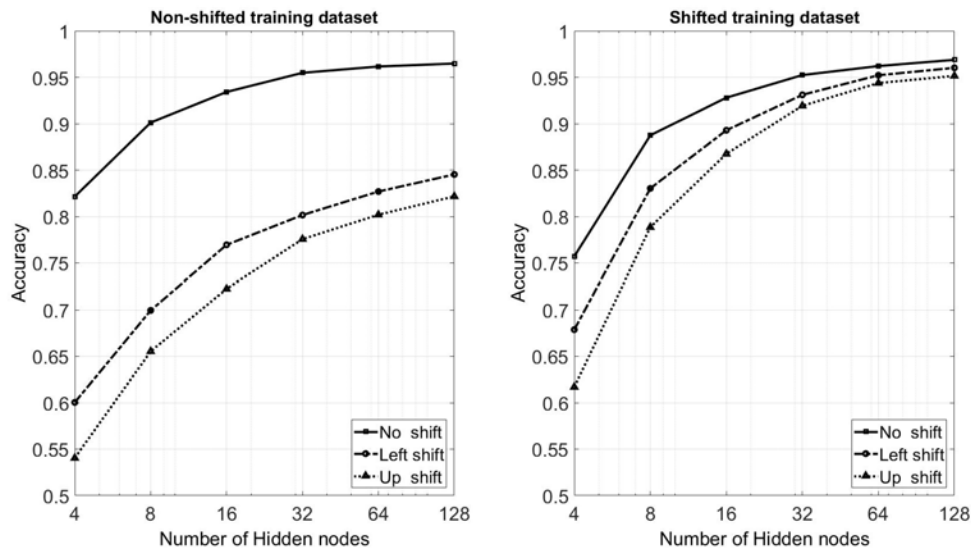
Στο στάδιο της προσομοίωσης και της ανάλυσης των παραμέτρων του συστήματος θεωρήθηκε σημαντικό να μελετηθεί η επίδραση του ίδιου του training Dataset στην ακρίβεια που επιτυγχάνεται από τον αλγόριθμο του classification ([18]). Ουσιαστικά η επιρροή του training Dataset εκφράζεται μέσα από τις μεταβολές στα προϊόντα του training, τα οποία αποτελούν τα weights και biases του RBM νευρωνικού δικτύου. Όπως εξηγήθηκε προηγούμενα, η παρούσα διπλωματική δεν έχει ως αντικείμενο τη διαδικασία του training, καθώς περιορίζεται στην επιτάχυνση του classification. Για το λόγο αυτό αξιοποιήθηκε κώδικας σε γλώσσα προγραμματισμού *Python* ο οποίος υλοποιεί το τμήμα του training και είναι υπεύθυνος για την παραγωγή των weights και biases σε μορφή πινάκων [18]. Ο κώδικας αυτός αποτελεί προϊόν προγενέστερης δουλειάς και δεν ανήκει στον συγγραφέα του παρόντος κειμένου.

Για τη μελέτη της επίδρασης του training Dataset αναπτύχθηκαν 5 νέα *Datasets*, με βάση το *reduced MNIST Dataset* (χάρην απλότητας θεωρούμε, από αυτό το σημείο της ανάλυσης και πέρα, ότι τα *Training* και *Test Datasets* αποτελούν ένα ενιαίο σύνολο, εκτός αν δηλώνεται ξεκάθαρα ο διαχωρισμός τους). Η λογική της μετατροπής των *Datasets* βασίζεται στην ιδέα της *ολίσθησης της εικόνας*. Αυτό σε τεχνικό επίπεδο μεταφράζεται σε ολίσθηση των pixels της εικόνας σε όλες τις δυνατές κατευθύνσεις και συγκεκριμένα *πάνω, κάτω, δεξιά και αριστερά*. Για την εν λόγω μελέτη αναπτύχθηκαν 4 νέα *Datasets*, όλα αντίγραφα του *reduced MNIST Dataset*, με τη διαφορά ότι κάθε ένα περιέχει εικόνες, κατά ένα pixel ολισθημένες προς μία συγκεκριμένη κατεύθυνση. Αυτό σημαίνει ότι πλέον υπάρχουν: το αρχικό *reduced MNIST*

Dataset, το *left-shifted reduced MNIST Dataset*, το *right-shifted reduced MNIST Dataset*, το *up-shifted reduced MNIST Dataset* και το *down-shifted reduced MNIST Dataset*. Το πέμπτο νέο Dataset το οποίο δημιουργήθηκε, αποτελεί ένα υπερσύνολο των 5 προηγούμενων Datasets, καθώς περιέχει όλες τις εικόνες από κάθε ένα από αυτά, διατεταγμένες σε τυχαία σειρά. Το τελευταίο ονομάζεται *Super-shifted reduced MNIST Dataset* και έχει πενταπλάσιο μέγεθος από τα προηγούμενα σε αριθμό εικόνων. Η ανάλυση βασίστηκε στα παρακάτω βήματα:

1. Δημιουργία 4 Shifted Datasets και ενός υπερσυνόλου αυτών, το *Super-shifted reduced MNIST Dataset*. Αυτά διαχωρίζονται για τις λειτουργίες του training και του classification και άρα υπάρχει ένα Training ένα Test Dataset για κάθε μία περίπτωση.
2. Εκτέλεση του αλγορίθμου του *training* με χρήση του αρχικού *reduced, training MNIST Dataset*.
3. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *no-shift, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας που προκύπτει.
4. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *left-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
5. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *right-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
6. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *up-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
7. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *down-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
8. Εκτέλεση του αλγορίθμου του *training* με χρήση του *Super-shifted, reduced, training MNIST Dataset*.
9. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *no-shift, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
10. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *left-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
11. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *right-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
12. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *up-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
13. Εκτέλεση του αλγορίθμου του *classification* με χρήση του *down-shifted, reduced, test MNIST Dataset*. Μέτρηση και καταγραφή της ακρίβειας.
14. Σύγκριση των μερήσεων ακρίβειας και παρουσίαση των αποτελεσμάτων σε γραφική παράσταση.

Το σχήμα 4.6 παρουσιάζει τα διαγράμματα της ακρίβειας για τις διαφορετικές περιπτώσεις των Datasets. Στο αριστερό σχήμα φαίνονται τα αποτελέσματα των βημάτων 2 – 6, ενώ στο δεξιό τα αντίστοιχα των βημάτων 8 – 13. Καθώς ήταν αναμενόμενο, η περίπτωση του training με χρήση του *Super-shifted, reduced, training MNIST Dataset* δίνει καλύτερα αποτελέσματα από πλευράς ακρίβειας. Ωστόσο, παρατηρείται ότι για μικρότερο πλήθος από hidden nodes, η ακρίβεια του αλγορίθμου με τη χρήση του αρχικού *no-shift, reduced, test MNIST Dataset* είναι χειρότερη, για το ίδιο training dataset. Αυτό εξηγείται από το γεγονός ότι, ένα δίκτυο RBM μπορεί να μοντελοποιήσει λιγότερες εξαρτήσεις, όταν περιλαμβάνει μικρότερο αριθμό από hidden nodes. Υπενθυμίζεται ότι hidden nodes λειτουργούν ως *feature detectors*. Για το λόγο αυτό παρουσιάζεται το φαινόμενο, το δίκτυο να εμφανίζει υψηλή ακρίβεια για τα δεδομένα εισόδου της διαδικασίας του training, αλλά ταυτόχρονα χαμηλή ικανότητα ταυτοποίησης στη διαδικασία του classification, όπου χρησιμοποιούνται διαφορετικά δεδομένα εισόδου (Test Datasets). Το φαινόμενο αυτό αναφέρεται επιγραμματικά ως *Overfitting*. Στο σχήμα 4.6 παρουσιάζονται μόνο οι περιπτώσεις των *left-shifted* και *up-shifted* καθώς οι υπόλοιπες είναι αντίστοιχες σε ακρίβεια.

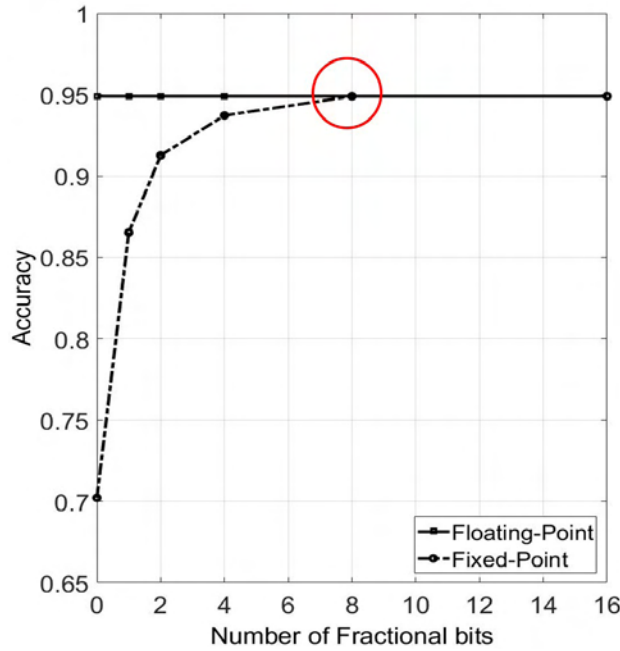


Σχήμα 4.6: Η επίδραση των διαφορετικών training datasets στην ακρίβεια του Classification

4.4 Hardware Accelerator: Σχεδιαστικές επιλογές

4.4.1 Αριθμητική Υπολογισμών

Τελευταίο σκέλος κατά τη διαδικασία του σχεδιασμού και της ανάλυσης του HDR-RBM Hardware Accelerator αποτέλεσε η επιλογή της αριθμητικής των πράξεων που θα υλοποιηθούν σε επίπεδο υλικού στο FPGA. Οι δυνατές επιλογές περιορίστηκαν σε πράξεις *fixed* και *floating point αριθμητικής*. Ο περιορισμός αυτός εισήχθη από την ίδια τη φύση των αριθμών που χρησιμοποιούνται ως δεδομένα στον υπολογισμό, αλλά και των πράξεων προς υλοποίηση. Τα weights και biases με τα οποία τροφοδοτείται η διαδικασία του classification είναι πραγματικοί αριθμοί.

Σχήμα 4.7: *RBM Accuracy vs Number of Fractional Bits*

Η λεπτομερής παρατήρηση των εξισώσεων του classification 3.4 που πρέπει να υλοποιηθούν οδηγεί στο συμπέρασμα ότι περιλαμβάνονται τόσο γραμμικοί, όσο και μη-γραμμικοί υπολογισμοί. Συγκεκριμένα οι πράξεις προς υλοποίηση περιλαμβάνουν: πρόσθεση, αφαίρεση, πολλαπλασιασμό και σύγκριση, οι οποίες είναι γραμμικές συναρτήσεις, αλλά και λογάριθμο και εκθετικό, που είναι μη-γραμμικές συναρτήσεις. Για τις δύο αυτές μη γραμμικές συναρτήσεις έγινε η σχεδιαστική επιλογή να υλοποιηθούν με τη χρήση floating point αριθμητικής. Οι λόγοι που οδήγησαν σε αυτή την επιλογή θα φωτιστούν με λεπτομέρεια στο κεφάλαιο 5 στο οποίο θα περιγραφεί αναλυτικά η αρχιτεκτονική και η υλοποίηση του HDR-RBM Hardware Accelerator. Όσον αφορά τις γραμμικές πράξεις ήταν θεμιτό να εξεταστεί αν αυτές μπορούν να υλοποιηθούν με τη χρήση fixed point αριθμητικής. Οι λόγοι αυτής της προσπάθειας είναι προφανείς και στηρίζονται στα πλεονεκτήματα που παρουσιάζει η fixed point αριθμητική από τη σκοπιά του latency των πράξεων, αλλά και των απαιτούμενων πόρων, με όρους σχεδίασης σε επίπεδο FPGA.

Η ανάλυση για το κατά πόσο είναι δυνατή η χρήση fixed point αριθμητικής ακολούθησε δύο κατευθύνσεις. Αρχικά, μέσω του μοντέλου προσομοίωσης υπολογίστηκε ο μέγιστος ακέραιος αριθμός που χρειαζόταν να αναπαρασταθεί. Βρέθηκε ότι 8 bits αρκούν για την αναπαράσταση κάθε δυνατού αποτελέσματος, καθώς για κάθε ένα από αυτά ισχύει η ανισότητα: $-128 \leq x \leq 127$. Άρα χρησιμοποιούνται 7 bits για την τιμή και ένα bit για το πρόσημο. Η δεύτερη κατεύθυνση αφορά τον αριθμό των δεκαδικών ψηφίων που είναι απαραίτητος για την διατήρηση της ακρίβειας του αλγορίθμου. Με αυτόν τον τρόπο εξασφαλίζεται ότι δεν εισαγάγεται σφάλμα στην εκτέλεση των πράξεων, λόγω της χρήσης fixed point αριθμητικής. Η ανάλυση αυτή έδειξε ότι είναι δυνατή η χρήση fixed point αριθμητικής χωρίς ουσιαστικό κόστος στην ακρίβεια των πράξεων. Όπως παρουσιάζεται στο σχήμα 4.7, ο απαραίτητος αριθμός δεκαδικών ψηφίων που εγγυάται μηδενικό σφάλμα σε επίπεδο αποτελεσμάτων είναι 8 bits. Στη γραφική αυτή παράσταση φαίνεται ότι η περαιτέρω αύξηση του αριθμού των

δεκαδικών ψηφίων δεν προσφέρει στην ακρίβεια του συστήματος, καθώς αυτή έχει την τιμή που επιτυγχάνεται με τη χρήση floating point αριθμητικής για το σύνολο των πράξεων.

Συνοψίζοντας, λοιπόν, επιλέχθηκε η χρήση floating point αριθμητικής για την υλοποίηση μη-γραμμικών πράξεων και fixed point αριθμητικής 16 *bits* δεκαδικών αριθμών με 8 *bits* ακέραιο μέρος και 8 *bits* δεκαδικό, για την υλοποίηση όλων των γραμμικών πράξεων. Περισσότερα αναφορικά με τις δύο αυτές επιλογές θα παρουσιαστούν στο κεφάλαιο 5.

Κεφάλαιο 5

Αρχιτεκτονική του Hardware Accelerator

Στόχος της παρούσας διπλωματικής εργασίας ήταν η σχεδίαση και η υλοποίηση μίας βιβλιοθήκης που περιλαμβάνει έναν RBM Hardware accelerator, ο οποίος επιλύει το πρόβλημα της αυτόματης αναγνώρισης χειρόγραφων ψηφίων ([19],[20],[18]). Όπως αναλύθηκε στο κεφάλαιο 4, ο Hardware Accelerator που περιγράφεται υλοποιεί τη διαδικασία του Classification με τη χρήση ενός νευρωνικού δικτύου RBM. Το αποτέλεσμα της διαδικασίας αυτής είναι η ταυτοποίηση του ψηφίου που αντιπροσωπεύει η εικόνα που χρησιμοποιείται ως είσοδος. Οι εξισώσεις που περιγράφουν τις διαδικασίες του Classification παρουσιάστηκαν στο κεφάλαιο 3 και έχουν χωριστεί σε 3 διακριτές υπολογιστικές φάσεις. Κάθε μία εκτελεί πεπερασμένο αριθμό πράξεων για τον υπολογισμό της συνεισφοράς κάθε visible και hidden node στην *free energy* κάθε πιθανού ψηφίου. Παρακάτω θα αναλυθούν εκτενώς οι φάσεις υπολογισμού καθώς και η συνολική αρχιτεκτονική του συστήματος του accelerator.

5.1 Εξισώσεις Classification και Φάσεις υπολογισμού

Με βάση τις εισερχόμενες εικόνες, υπολογίζονται τα weights και τα biases του δικτύου κατά τη διαδικασία του training. Οι παρακάτω εξισώσεις εφαρμόζονται για την περίπτωση του Classification:

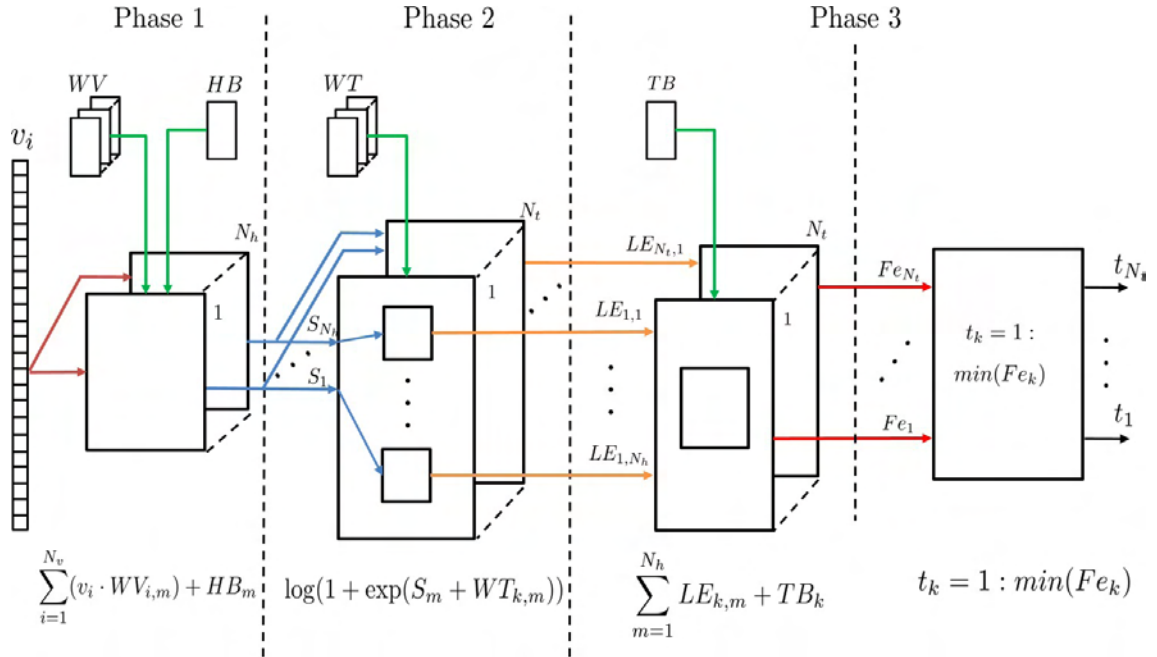
$$S_m = \sum_{i=1}^{N_v} (v_i \cdot WV_{i,m}) + HB_m \quad (5.1)$$

$$LE_{k,m} = \ln(1 + \exp(S_m + WT_{k,m})) \quad (5.2)$$

$$Fe_k = -(\sum_{m=1}^{N_h} LE_{k,m} + TB_k) \quad (5.3)$$

$$t_k = 1 : \min(Fe_k) \quad (5.4)$$

όπου $m \in [1, N_h]$, $k \in [1, N_t]$. Ως v_i συμβολίζεται ο visible node i , ως h_m συμβολίζεται ο hidden node m και t_k είναι ο target node k . Fe_k είναι η *free energy* του target node k . Τα weights και τα biases ορίζονται παρακάτω: VB_i είναι τα biases των visible nodes, TB_k είναι τα biases των target nodes, HB_m είναι τα biases των hidden nodes, WV είναι τα weights ανάμεσα στους visible και hidden nodes, ενώ WT είναι τα weights ανάμεσα στους target και hidden nodes.



Σχήμα 5.1: Αρχιτεκτονική RBM Hardware Accelerator

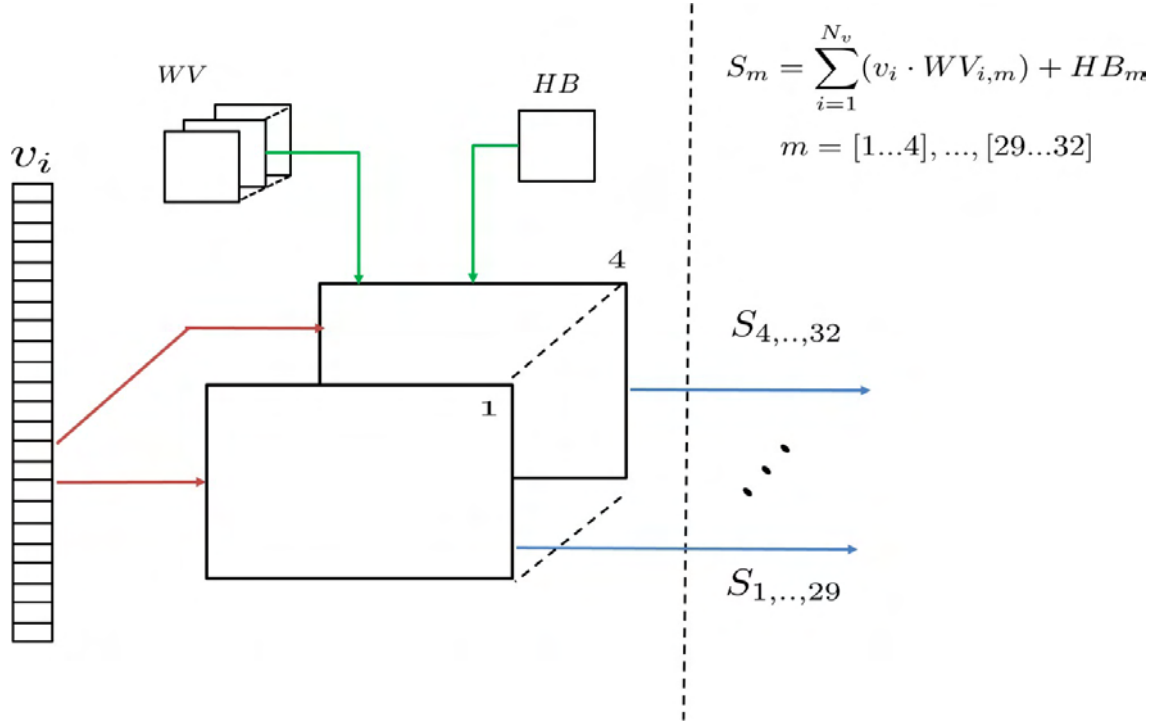
Οι εξισώσεις που παρουσιάζονται στο σημείο αυτό είναι ταυτόσημες με αυτές στο κεφάλαιο 3, 3.4, με άλλη όμως μορφή. Σκοπός της μετατροπής αυτής είναι ο σαφής διαχωρισμός των απαιτούμενων πράξεων και η κατανομή τους σε τρεις (3) διακριτές φάσεις.

Η εξίσωση 5.1 αντιπροσωπεύει τον υπολογισμό της συνεισφοράς των visible nodes στην free energy κάθε ψηφίου. Αντίστοιχα η εξίσωση 5.2 υπολογίζει τη συνεισφορά των hidden nodes και η εξίσωση 5.3 είναι υπεύθυνη για τον καθαυτό υπολογισμό της free energy κάθε πιθανού ψηφίου (από το 0 μέχρι το 9). Τέλος η εξίσωση 5.4 δείχνει την πράξη της σύγκρισης ανάμεσα στις δέκα free energies για την επιλογή του ψηφίου με τη μικρότερη, ως αυτού που τελικά αναγνωρίζεται.

Η εξίσωση 5.1 περιγράφει τις πράξεις που υλοποιούνται από την υπολογιστική φάση ένα (1), η 5.2 υλοποιείται από την υπολογιστική φάση δύο (2), ενώ οι εξισώσεις 5.3 και 5.4 συνδυασμένες, αποτελούν την υπολογιστική φάση τρία (3).

5.2 Hardware Accelerator: Δομικό Διάγραμμα

Η αρχιτεκτονική που παρουσιάζεται στο σχήμα 5.1 είναι γενική και περιγράφει όλες τις δυνατές υλοποιήσεις από τελείως παράλληλη και κατανεμημένη λογική μέχρι σειριακή. Στο δομικό αυτό διάγραμμα φαίνονται οι διεπαφές ανάμεσα στις υπολογιστικές φάσεις, καθώς και η μέθοδος παραλληλοποίησης που μπορεί να εφαρμοστεί. Βασική κατεύθυνση κατά το στάδιο της σχεδίασης του accelerator αποτέλεσε ο βέλτιστος συγκεκριμένος: πλήθους δομικών στοιχείων - μέγιστου δυνατού ρυθμού επεξεργασίας εικόνων. Η βιβλιοθήκη του συγκεκριμένου RBM accelerator σχεδιάστηκε με τέτοιο τρόπο, ώστε να προσαρμόζεται με ευκολία στις απαιτήσεις υλοποίησης. Πιο ειδικά, η γενικευμένη σχεδίαση που παρουσιάζεται στο σχήμα 5.1 είναι παραμετροποιήσιμη αναφορικά με τον αριθμό των μονάδων που θα

Σχήμα 5.2: Αρχιτεκτονική 1^{ης} Υπολογιστικής Φάσης

λειτουργούν παράλληλα και ανά φάση. Οι τιμές των παραμέτρων αυτών εξαρτώνται άμεσα από το μέγεθος και την γενιά του FPGA το οποίο χρησιμοποιείται για την υλοποίηση. Έτσι, ο βαθμός παραλληλοποίησης αποτελεί μία σχεδιαστική επιλογή στο επίπεδο της τελικής ενσωμάτωσης του accelerator στο συνολικό σύστημα. Στο κεφάλαιο αυτό θα παρουσιαστεί μία συγκεκριμένη υλοποίηση που σχετίζεται με τις δυνατότητες του FPGA board το οποίο χρησιμοποιήθηκε και αυτό ήταν το *ADM-PCIE-7V3* [21]. Ωστόσο, με βάση τη γενικότητα της αρχιτεκτονικής του σχήματος 5.1 ο τελικός accelerator μπορεί να τροποποιηθεί με ευκολία και να ανταποκριθεί στις δυνατότητες ενός μεγαλύτερου ή μικρότερου FPGA.

Τα επόμενα υποτιμήματα είναι αφιερωμένα στην αναλυτική περιγραφή κάθε υπολογιστικής φάσης ξεχωριστά. Θα παρουσιαστούν οι σχεδιαστικές επιλογές, στοιχεία που αφορούν χρονισμούς, καθώς και απαιτούμενους πόρους για το συγκεκριμένο FPGA στο οποίο βασίστηκε η υλοποίηση [21].

5.2.1 Πρώτη Υπολογιστική Φάση

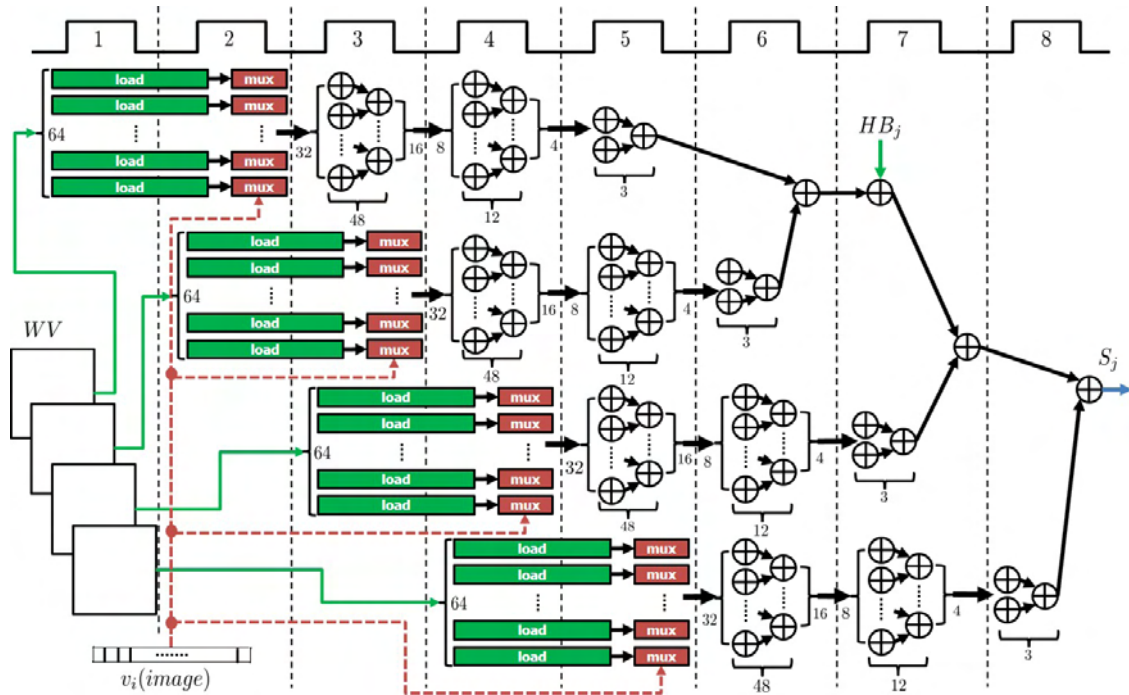
Η φάση 1 υπολογίζει το άθροισμα των weights των visible nodes που συνεισφέρουν στην free energy κάθε ψηφίου. Τα αποτελέσματα που προκύπτουν από το κύκλωμα της φάσης 1 είναι κοινά για όλα τα ψηφία και για το λόγο αυτό χρειάζεται να υπολογιστούν μία φορά για κάθε εικόνα. Όπως φαίνεται από την εξίσωση του σχήματος 5.2, οι βασικές προς υλοποίηση συναρτήσεις είναι αυτή της πρόσθεσης και του πολλαπλασιασμού. Σε επίπεδο υλικού, η πράξη του πολλαπλασιασμού είναι αρκετά κοστοβόρα, καθώς έχει υψηλές απαιτήσεις τόσο σε πόρους, όσο και *latency*. Με βάση τις προδιαγραφές του *RBM* νευρωνικού δικτύου, οι visible nodes αποτελούν δυαδικά στοιχεία, παίρνουν δηλαδή τιμές μηδέν (0) ή ένα (1). Κάνοντας χρήση της συγκεκριμένης προδιαγραφής, μπορεί πολύ ευκολα η πράξη του πολλαπλασιασμού

Function	BRAM18KB	DS P48E	FF	LUT
multiplication	0	1	72	34
multiplexing	0	0	0	24

Πίνακας 5.1: Απαιτούμενοι πόροι πολλαπλασιασμού - πολύπλεξης (Virtex - 7)

να μετατραπεί σε πολύπλεξη. Αυτό απλοποιεί αρκετά την πολυπλοκότητα της 1^{ης} φάσης, ενώ επιταχύνει ακόμα περισσότερο τους υπολογισμούς για τη συνεισφορά των visible nodes στην τελική ενέργεια. Λεπτομέρειες σχετικές με τους απαιτούμενους πόρους για τις δύο συναρτήσεις, συγκριτικά, φαίνονται στον πίνακα 5.1.

Στο δομικό διάγραμμα του σχήματος 5.2 φαίνονται τα στοιχεία που συνθέτουν την 1^η φάση. Είσοδοι της φάσης αυτής αποτελούν τα pixel της εκάστοτε εικόνας, τα οποία, σε επίπεδο αλγορίθμου, αφορούν τους visible nodes. Από το σχήμα 5.2 γίνεται κατανοητό ότι, από τη στιγμή που χρησιμοποιούνται 32 hidden nodes, πρέπει να υπολογιστούν 32 S_m αποτελέσματα. Τα αποτελέσματα αυτά θα μπορούσαν να υπολογίζονται με διαφορετικούς βαθμούς παραλληλοποίησης, ανάλογα με τις προδιαγραφές υλοποίησης, με δυνατότητα να προκύπτουν από 1 έως 32 αποτελέσματα ταυτόχρονα. Για τη φάση αυτή έγινε η επιλογή να υλοποιηθούν 4 παράλληλες μονάδες, όπου κάθε μία είναι υπεύθυνη για τον υπολογισμό 8 S_m αποτελεσμάτων. Η επιλογή των 4 παράλληλων μονάδων αποτελεί ένα συμβιβασμό ανάμεσα στο μέγιστο επιτρεπόμενο latency για τη φάση 1 και στον ελάχιστο δυνατό αριθμό απαιτούμενων πόρων και αφορά τη συγκεκριμένη υλοποίηση με βάση το ADM-PCIE-7V3 FPGA board.



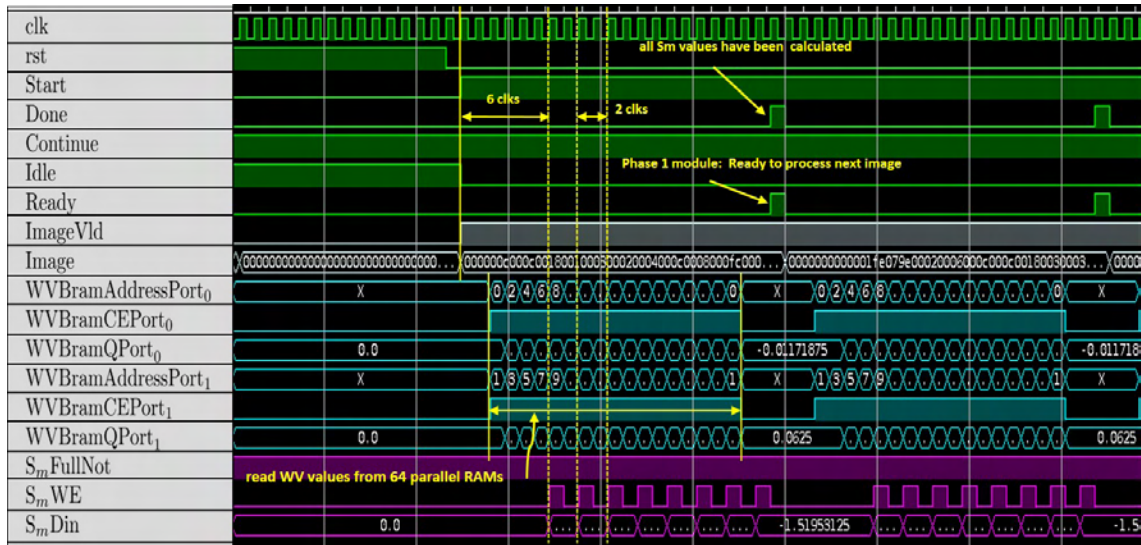
Σχήμα 5.3: Υπολογιστική Φάση 1: Χρονικό Διάγραμμα

Στο σχήμα 5.3 φαίνονται αναλυτικά, τόσο το σύνολο των πράξεων που πρέπει να υλοποιηθούν από κάθε δομική μονάδα, όσο και ο χρονοπρογραμματισμός των πράξεων αυτών. Για

τη βέλτιστη εφαρμογή της τεχνικής του pipeline, εσωτερικά της 1^{ης} φάσης, έγινε η επιλογή να χρησιμοποιηθούν 256 αθροιστές, με δεδομένο ότι αυτός είναι και ο αριθμός των απαιτούμενων αθροίσεων για τον υπολογισμό ενός S_m αποτελέσματος. Οι προσθέσεις εκτελούνται σε μορφή *adder trees*, όπου, ανά κύκλο ρολογιού, κάθε βήμα τροφοδοτεί το αποτέλεσμα του στο επόμενο, όπως φαίνεται στο διάγραμμα 5.3. Με αυτόν τον τρόπο και με τη χρήση της τεχνικής του pipeline, κάθε παράλληλη μονάδα επιτυγχάνει latency 21 κύκλων ρολογιού. Όπως θα δειχθεί και στη συνέχεια, η συγκεκριμένη απόδοση καλύπτει τις προδιαγραφές σε latency και πόρους συστήματος.

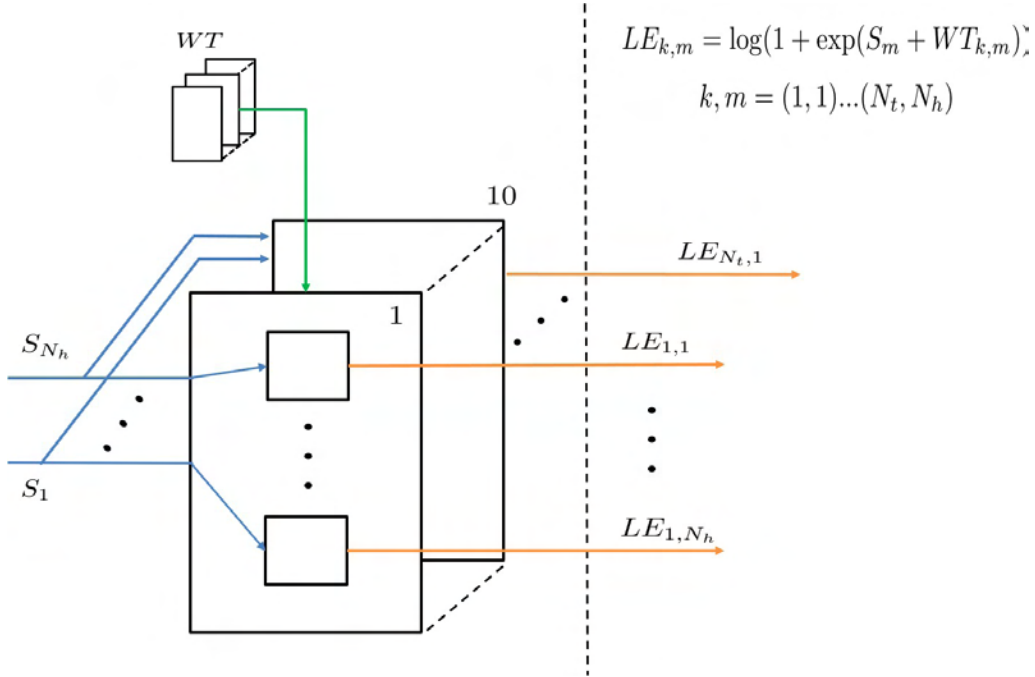
Οι πράξεις της πολύπλεξης και της πρόσθεσης είναι γραμμικές και άρα με βάση την ανάλυση που προηγήθηκε στο κεφάλαιο 3 υλοποιήθηκαν, σε επίπεδο υλικού, με τη χρήση fixed-point αριθμητικής 16-bits αριθμών.

Για τις πράξεις της 1^{ης} φάσης είναι απαραίτητη η ύπαρξη δύο πινάκων που περιλαμβάνουν τα προυπολογισμένα από τη διαδικασία του training weights και biases των visible nodes. Από την πλευρά της υλοποίησης, οι πίνακες αυτοί μεταφράζονται σε FPGA - Block RAMs δομικά στοιχεία. Με δεδομένο ότι κάθε weight είναι ένας 16-bits αριθμός και ότι με βάση το χρονοπρογραμματισμό φορτώνονται 64 weights παράλληλα (σχήμα 5.3), χρησιμοποιήθηκαν παράλληλα 128 *BRAMs* μεγέθους 18KB, με πλάτος 8-bit η κάθε μία.

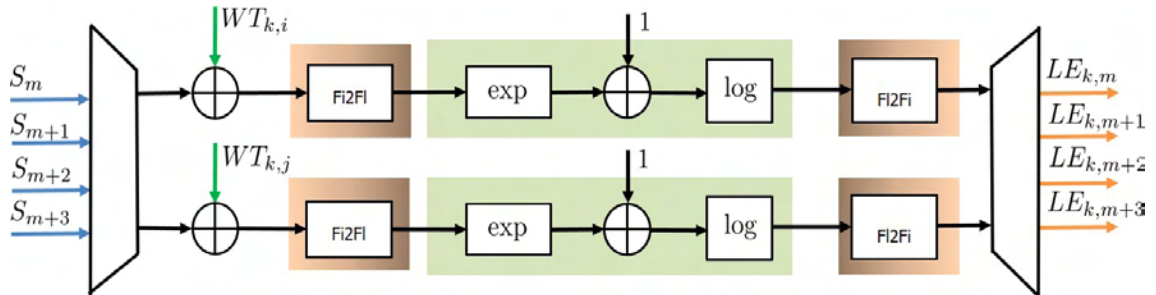


Σχήμα 5.4: Υπολογιστική Φάση 1: Timing Waveform

Στην εικόνα 5.4 παρουσιάζονται με αναλυτικό τρόπο οι διαδικασίες της φάσης 1, από τη σκοπία των χρονισμών. Φαίνεται ότι το πρώτο S_m αποτέλεσμα είναι διαθέσιμο 6 κύκλους μετά από τη στιγμή που ο accelerator τροφοδοτείται με την εικόνα εισόδου. Ακόμα από τη συγκεκριμένη κυματομορφή γίνεται αντιληπτό ότι, λόγω της τεχνικής του pipeline, όλες οι S_m έξοδοι της υπολογιστικής φάσης 1 έχουν παραχθεί 20 κύκλους ρολογιού μετά την εκκίνηση. Αυτό συμβαίνει γιατί εξασφαλίζεται συνεχής τροφοδότηση και λειτουργία των adder trees με βάρη, ανα δύο κύκλους, μέχρι να ολοκληρωθεί ο απαιτούμενος αριθμός προσθέσεων (256 ανά S_m). Τα παραπάνω σημαίνουν ότι η 1^η φάση μπορεί να λαμβάνει μία νέα εικόνα ανά 20 κύκλους και άρα με συχνότητα ρολογιού 125 MHz επιτυγχάνεται *processing rate* της τάξης του 6.25 MImages/s.

Σχήμα 5.5: Αρχιτεκτονική 2^{ης} Υπολογιστικής Φάσης

5.2.2 Δεύτερη Υπολογιστική Φάση

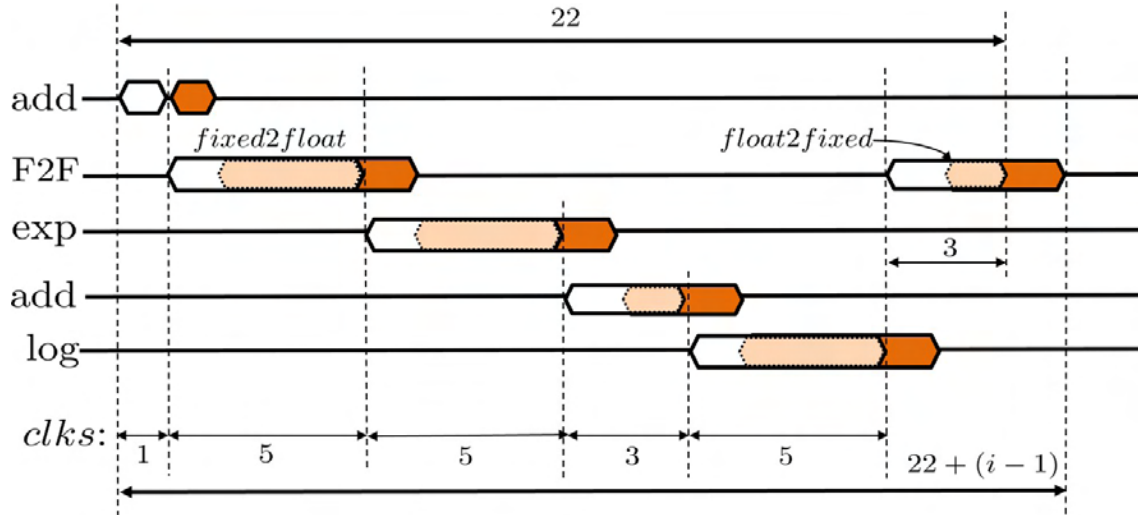


Σχήμα 5.6: Υπολογιστική Φάση 2: Εσωτερική Αρχιτεκτονική

Η 2^η φάση είναι υπεύθυνη για τον υπολογισμό της πιθανότητας κάθε hidden node να πάρει την τιμή 0 ή 1, βασισμένη στα αποτελέσματα της 1^{ης} υπολογιστικής φάσης. Ουσιαστικά υπολογίζεται ο λογάριθμος της προαναφερθείσας πιθανότητας, μιάς και αυτό είναι αρκετό για την τελική σύγκριση των free energies των ψηφίων. Το αποτέλεσμα αυτό τροφοδοτείται στη συνέχεια στο κύκλωμα της φάσης 3 για τον τελικό υπολογισμό της συνολικής ενέργειας κάθε ψηφίου. Στο σχήμα 5.5 παρουσιάζεται η διάταξη των δομικών στοιχείων της φάσης 2 και η λογική με την οποία η φάση αυτή έχει παραλληλοποιηθεί και υλοποιηθεί.

Ο αριθμός των hidden nodes είναι 32 και άρα τόσα είναι τα αποτελέσματα που πρέπει να προκύψουν για κάθε target node, δηλαδή για κάθε ψηφίο. Με δεδομένο ότι ο αριθμός των ψηφίων είναι 10, τελικά πρέπει να υπολογιστούν $10 \times 32 = 320$ αποτελέσματα. Θα ακολουθήσει μια ανάλυση σχετικά με την τελική υλοποίηση του κυκλώματος της 2^{ης} φάσης και τις σχεδιαστικές επιλογές που οδήγησαν σε αυτή.

Τα σχήματα 5.5, 5.6, 5.7 και 5.8 είναι αφιερωμένα στη 2^η φάση. Βασικές συναρτήσεις για



Σχήμα 5.7: Υπολογιστική Φάση 2: Χρονικό Διάγραμμα

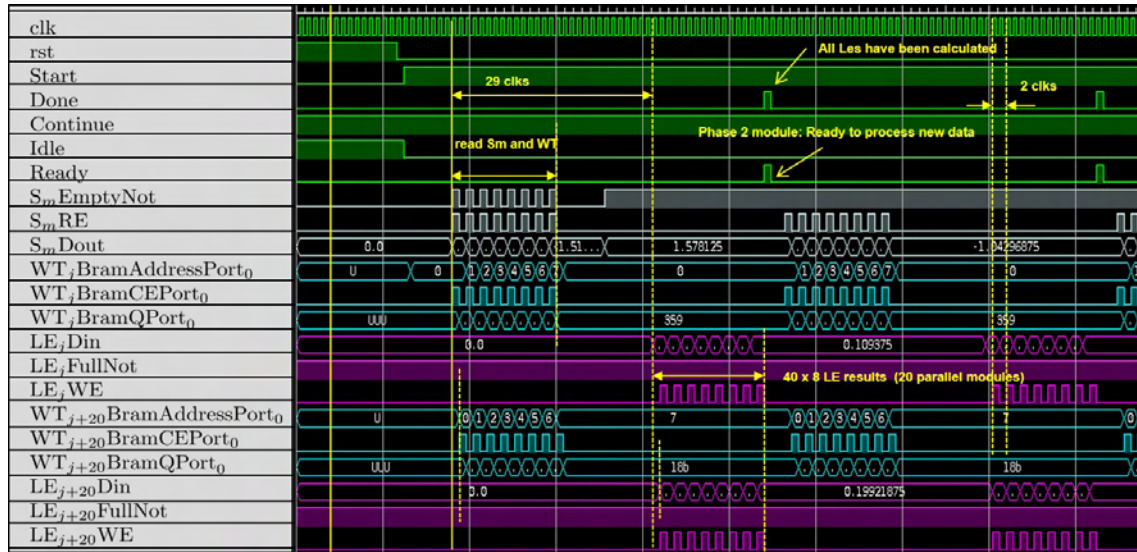
τη φάση αυτή αποτελούν η **εκθετική** και η **λογαριθμική**. Η περιγραφή που προηγήθηκε στο κεφάλαιο 4 αναφέρει ότι για όλες τις μη γραμμικές συναρτήσεις γίνεται χρήση floating-point αριθμητικής. Βασικός λόγος που οδήγησε σε αυτή την επιλογή ήταν χρήση της βιβλιοθήκης *Floating-Point Operator* της Xilinx ([22]), η οποία για τις μη γραμμικές συναρτήσεις έχει σαν προϋπόθεση τη χρήση floating point αριθμητικής. Επιπρόσθετα στη φάση αυτή, πρέπει να υπολογιστούν δύο αθροίσεις. Στο σχήμα 5.6 παρουσιάζεται η εσωτερική αρχιτεκτονική καθώς και στο σχήμα 5.7 η εφαρμογή του pipeline εσωτερικά των πράξεων.

Όσον αφορά την αρχιτεκτονική, από το δομικό διάγραμμα φαίνεται ότι μετά την πρώτη άθροιση ακολουθεί μία μονάδα *Fixed to Floating (Fi2Fi)*, ενώ μετά τον υπολογισμό του λογαρίθμου προστίθεται μία μονάδα *Floating to Fixed (Fi2Fi)*. Οι μονάδες αυτές επιστρατεύτηκαν για λόγους συμβατότητας ανάμεσα σε fixed και floating point αριθμητική. Μία λεπτομέρεια στο σημείο αυτό αποτελεί η υλοποίηση της πράξης της πρόσθεσης της *μονάδας*, πριν τον υπολογισμό του λογαρίθμου, με χρήση floating point αριθμητικής. Λόγω του ότι η πράξη αυτή δεν έχει υψηλές απαιτήσεις ούτε σε πόρους, ούτε σε χρόνο εκτέλεσης, υλοποιήθηκε σε floating point αριθμητική, ώστε να αποφευχθεί η χρήση επιπλέον μονάδων συμβατότητας.

Στο χρονικό διάγραμμα του 5.7 δίνονται οι απαιτούμενοι χρόνοι εκτέλεσης για κάθε πράξη ξεχωριστά, αλλά και ο τρόπος που συνδυάζονται με τη μέθοδο του pipeline. Από τη χρονική στιγμή που το πρώτο S_m αποτέλεσμα γίνεται διαθέσιμο, η 2^η φάση χρειάζεται 28 κύκλους ρολογιού για να παράξει το πρώτο $LE_{k,m}$ αποτέλεσμα.

Ιδανικά όλα τα $LE_{k,m}$ αποτελέσματα θα μπορούσαν να υπολογίζονται παράλληλα, ώστε να επιτυγχάνεται ο μέγιστος δυνατός ρυθμός επεξεργασίας. Η επιλογή αυτή, όμως, θα είχε μεγάλο αντίκτυπο στους απαιτούμενους πόρους του FPGA και άρα στο πόσα νευρωνικά δίκτυα θα μπορούσαν να ενσωματωθούν παράλληλα στο τελικό σύστημα. Όπως αναφέρθηκε προηγουμένως, βασική αρχή κατά το σχεδιασμό του accelerator είναι ο βέλτιστος συγκερασμός ελάχιστου latency και εκμετάλλευσης των πόρων του FPGA. Τα δύο βασικά στοιχεία που καθορίζουν τον βέλτιστο αριθμό παράλληλων μονάδων αποτελούν τα *DSP48E primitive resources*, καθώς και τα *LookUp Tables* ([23]). Οι δομικές μονάδες *DSP48E* είναι ειδικά σχεδιασμένες για τη γρήγορη υλοποίηση αριθμητικών πράξεων. Ο πίνακας 5.2 δίνει μία αίσθηση

για τους απαιτούμενους πορους σε διαφορετικές περιπτώσεις παραλληλοποίησης.



Σχήμα 5.8: Υπολογιστική Φάση 2: Timing Waveform

Με βάση τον πίνακα 5.2 και με δεδομένο το FPGA - Virtex 7, xc7vx690tffg1157-2, έγινε η επιλογή να χρησιμοποιηθούν 20 παράλληλες μονάδες υπολογισμού, ομαδοποιημένες ανά δύο, όπως ακριβώς δείχνει το σχήμα 5.6. Η ομαδοποίηση αυτή έγινε με στόχο κάθε προκύπτουσα δομή να είναι αφιερωμένη στους υπολογισμούς που αφορούν ένα ψηφίο. Η επιλογή αυτή έχει σαν αποτέλεσμα να δημιουργείται μία ασυμβατότητα ως προς το ρυθμό με τον οποίο παράγονται τα S_m αποτελέσματα από τη φάση 1 και αυτόν με το οποίο καταναλώνονται από τη φάση 2. Αυτό ισχύει γιατί, ενώ από τη φάση 1 παράγονται 4 S_m αποτελέσματα ταυτόχρονα, στη φάση 2 τροφοδοτούνται τα 2 από αυτά παράλληλα. Το πρόβλημα αυτό λύνεται με τη χρήση μίας μονάδας πολύπλεξης, η οποία φαίνεται στο σχήμα 5.6, στην είσοδο της κάθε υπολογιστικής δομής. Για τον αντίστοιχο λόγο, υπάρχει η αντίστροφη μονάδα στην έξοδο της φάσης 2. Η τελευταία επιτρέπει στη φάση 3 να τροφοδοτείται με 4 $LE_{k,m}$ αποτελέσματα παράλληλα όπως θα εξηγηθεί παρακάτω.

Κάθε υπολογιστική δομή της 2^{ης} φάσης, η αρχιτεκτονική της οποίας δίνεται από το σχήμα 5.6, είναι υπεύθυνη για τον υπολογισμό 16 $LE_{k,m}$ αποτελεσμάτων. Ακόμα από το σχήμα φαίνεται ότι πρέπει να φορτώνονται 2 $WT_{k,i}$ βάρη ανα κύκλο ρολογιού και με δεδομένο ότι κάθε weight είναι ένας 16 – bits fixed – point αριθμός, απαιτούνται 40 Distributed RAMs, με πλάτος 8-bit η κάθε μία. Η χρήση της Floating point βιβλιοθήκης δίνει τη δυνατότητα λήψης νέας εισόδου ανά κύκλο ρολογιού και άρα προσφέρει pipeline interval ίσο με 1. Με αυτόν τον τρόπο, από τη στιγμή που παράγεται το πρώτο αποτέλεσμα χρειάζονται ακριβώς 16 κύκλοι ρολογιού ακόμα, για να ολοκληρωθούν όλοι οι υπολογισμοί. Η 2^η φάση έχει συνολικό latency 47 κύκλους και μπορεί να δεχθεί νέα αποτελέσματα κάθε 48 κύκλους λόγω του pipeline. Έτσι επιτυγχάνεται ρυθμός επεξεργασίας μεγαλύτερος από 2.5 MImages/s ανά νευρωνικό δίκτυο, για τη φάση 2, με συχνότητα ρολογιού 125 MHz.

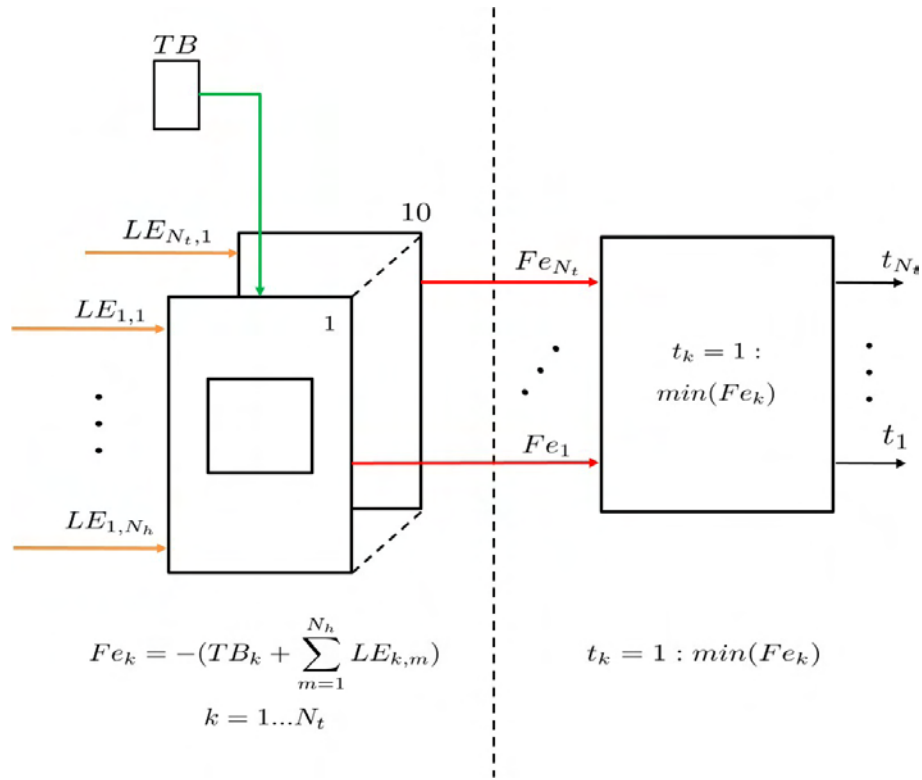
Η εικόνα 5.8 παρουσιάζει τις διαδικασίες για την ανάγνωση των εισόδων και των απαραίτητων παραμέτρων (όπως τα βάρη) καθώς και την αποθήκευση των αποτελεσμάτων της 2^{ης} φάσης. Η κυματομορφή αυτή δείχνει ότι το πρώτο LE αποτέλεσμα είναι διαθέσιμο 29

Parallel Units	5	20	40	80	Virtex7
DSP48E	110 – 6%	440 – 12%	880 – 24%	1760 – 24%	3600
LUT	40000 – 10%	90000 – 20%	150000 – 35%	270000 – 60%	433000

Πίνακας 5.2: Συγκριτικός πίνακας απαιτούμενων πόρων DSP48E και LUT για τη φάση 2 (Target device: Virtex 7, xc7vx690tffg1157-2)

κύκλους απο τη στιγμή που διαβάστηκε η πρώτη S_m είσοδος, ενώ μετά από συνολικά 47 κύκλους η φάση είναι έτοιμη να δεχθεί νέες εισόδους.

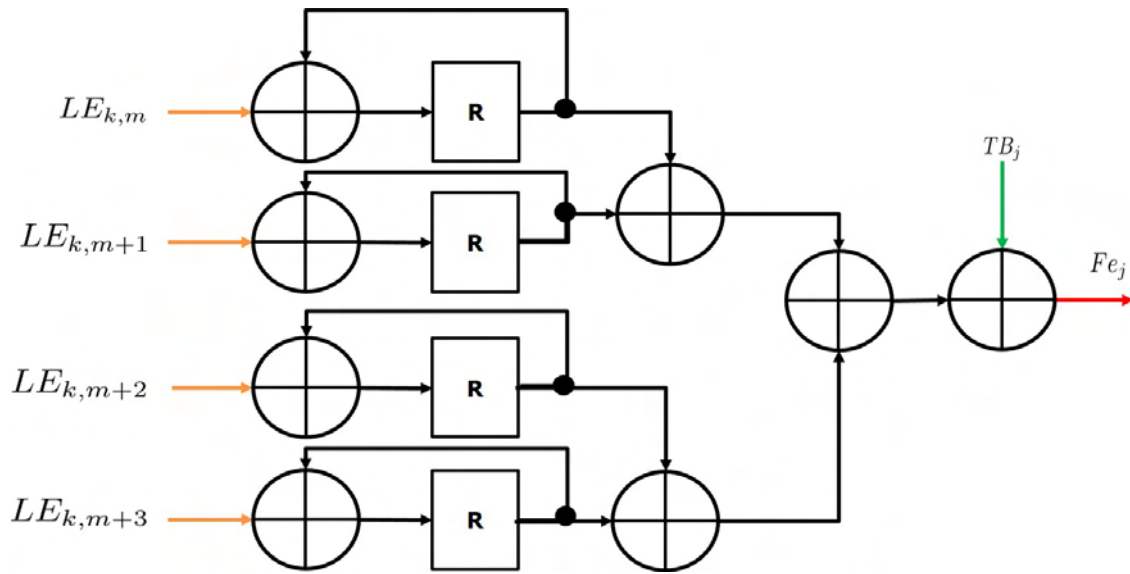
5.2.3 Τρίτη Υπολογιστική Φάση



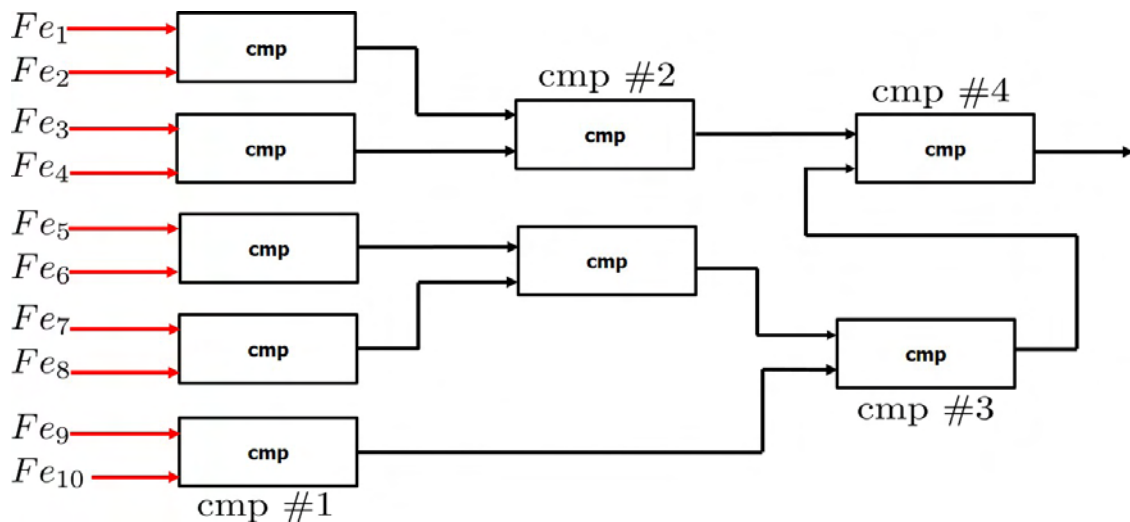
Σχήμα 5.9: Αρχιτεκτονική 3^{ης} Υπολογιστικής Φάσης

Η 3^η φάση είναι υπεύθυνη για τον υπολογισμό των free energies για όλα τα ψηφία, τη σύγκριση τους και τελικά, την αναγνώριση του χειρόγραφου ψηφίου. Η φάση χωρίζεται σε δύο τμήματα, όπως διακρίνεται από το σχήμα 5.9. Το πρώτο τμήμα περιλαμβάνει τα κυκλώματα που υπολογίζουν το τελικό άθροισμα και την εξαγωγή της free energy κάθε ψηφίου. Το δεύτερο τμήμα παίρνει σαν είσοδο το σύνολο των ενεργειών, τις συγκρίνει και βρίσκει το ψηφίο με την ελάχιστη. Στη φάση αυτή, όπως και στη φάση ένα, έγινε χρήση fixed point αριθμητικής, καθώς υλοποιούνται γραμμικές πράξεις.

Το πρώτο τμήμα της φάσης 3 ουσιαστικά υλοποιεί ένα συσσωρευτή (accumulator). Δέχεται σαν εισόδους τα LE αποτελέσματα από τη φάση 2 και τα προσθέτει. Ακόμα προστίθεται το bias του αντίστοιχου target node TB_k . Το δομικό διάγραμμα για το τμήμα αυτό δίνεται στο σχήμα 5.10. Οι fixed point adders απαιτούν 1 κύκλο για τον υπολογισμό ενός αποτελέσματος.

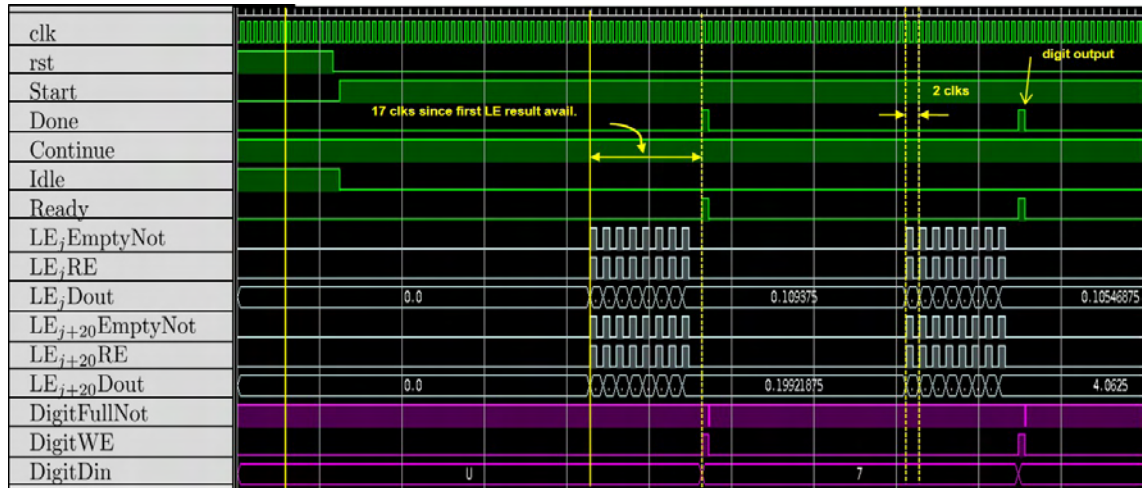
Σχήμα 5.10: Υπολογιστική Φάση 3: Αρχιτεκτονική 1^{ου} κυκλώματος

Η δομική μονάδα του σχήματος 5.10 τροφοδοτείται από τη φάση 2 με 4 παράλληλα LE αποτελέσματα από τα συνολικά 32, που είναι απαραίτητα για τον υπολογισμό μίας free energy. Τα παραπάνω έχουν σαν αποτέλεσμα κάθε δομική μονάδα του πρώτου τμήματος να απαιτεί χρόνο εκτέλεσης 11 κύκλων ρολογιού. Με δεδομένο ότι πρέπει να υπολογιστούν 10 free energies, όσες και ο αριθμός των target nodes, έγινε η επιλογή να χρησιμοποιηθούν 10 κυκλώματα ταυτόσημα με αυτό του σχήματος 5.10, παράλληλα. Λόγω της χρήσης fixed point αριθμητικής για τους αθροιστές, δεν υπάρχουν υψηλές απαιτήσεις σε πόρους και αυτό δίνει τη δυνατότητα μέγιστου βαθμού παραλληλοποίησης στο επίπεδο του πρώτου τμήματος της φάσης που εξετάζεται. Με αυτόν τον τρόπο, για τη φάση 3, επιτυγχάνεται latency 10 κύκλων ρολογιού για τον υπολογισμό του συνόλου των free energies.

Σχήμα 5.11: Υπολογιστική Φάση 3: Αρχιτεκτονική 2^{ου} κυκλώματος

Το δεύτερο τμήμα περιγράφεται από το σχήμα 5.11. Φαίνεται η αρχιτεκτονική σύμφωνα με την οποία υλοποιούνται οι συγκρίσεις και προκύπτει το τελικό αποτέλεσμα. Το συνολικό

latency του κυκλώματος αυτού είναι 4 κύκλοι, όπως γίνεται εύκολα κατανοητό από το συγκεκριμένο σχήμα 5.11 (1 κύκλος για κάθε παράλληλο συγκρητή). Συνολικά, η φάση 3 απαιτεί latency 17 κύκλων ρολογιού. Αυτό μεταφράζεται σε ρυθμό επεξεργασίας μεγαλύτερο από 6.2 *MImages/s* ανά νευρωνικό δίκτυο, για τη φάση 3, με τη χρήση ρολογιού συχνότητας 125 *MHz*.



Σχήμα 5.12: Υπολογιστική Φάση 3: Timing Waveform

Οι διαδικασίες ανάγνωσης των εισόδων και των παραμέτρων, από τη σκοπιά των χρονισμών παρουσιάζονται στις κυματομορφές του σχήματος 5.12. Εδώ φαίνεται ακόμα ο απαιτούμενος χρόνος για την τελική εξαγωγή του ψηφίου το οποίο αναγνωρίζεται.

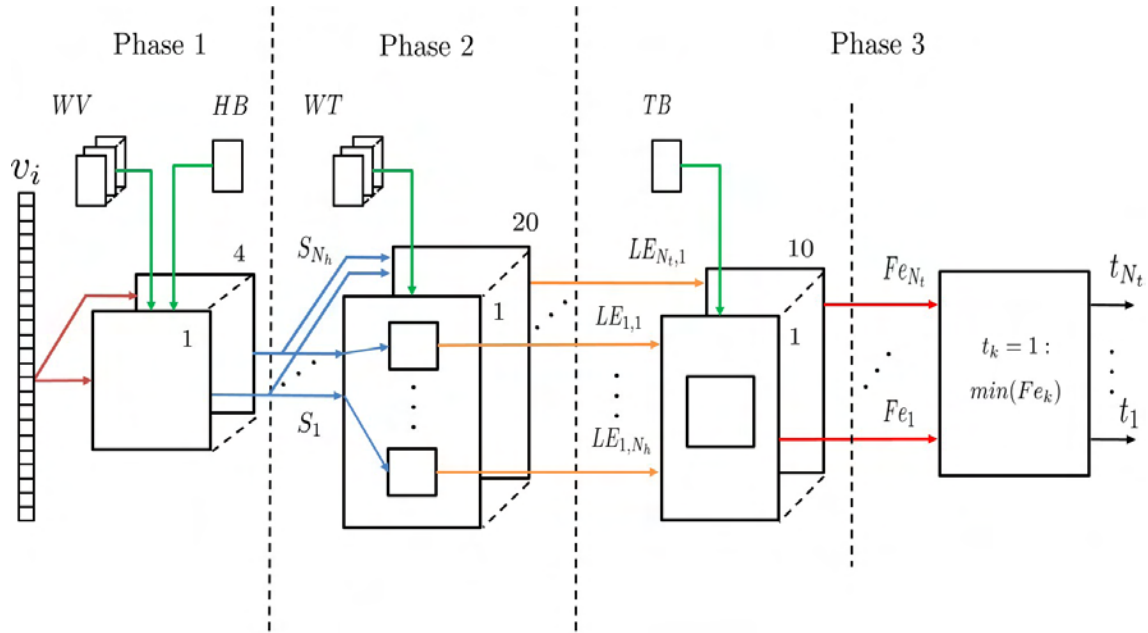
5.3 Εφαρμογή τεχνικών βελτιστοποίησης

Το δομικό διάγραμμα του σχήματος 5.13 αποτελεί μία συγκεκριμένη υλοποίηση της αρχιτεκτονικής που παρουσιάζεται στο 5.1. Η υλοποίηση αυτή έχει προκύψει μετά το συνδυασμό των κυκλωμάτων των φάσεων υπολογισμού που παρουσιάστηκαν, καθώς και των σχεδιαστικών επιλογών που τις ακολουθούν.

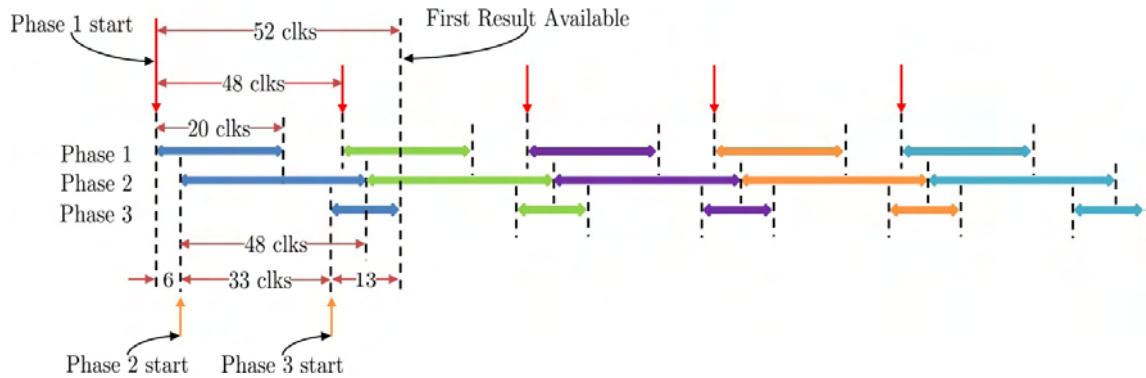
Όπως σε κάθε σύστημα που έχει σαν στόχο τη μέγιστη απόδοση, έτσι και στην παρούσα διπλωματική εργασία, αξιοποιήθηκαν τεχνικές βελτιστοποίησης σε επίπεδο υλικού. Βασική στόχευση αποτέλεσε η εξασφάλιση συνεχούς λειτουργίας του accelerator και των υποκυκλωμάτων που τον υλοποιούν. Συγκεκριμένα εφαρμόστηκαν η τεχνική του *ξεδιπλώματος* (*unfolding*) και φυσικά, όπως αναφέρθηκε, η τεχνική του *pipeline*.

Από την τελική αρχιτεκτονική που παρουσιάστηκε στο 5.13 είναι εμφανής η χρήση του *unfolding*. Ουσιαστικά αφορά τον βαθμό παραλληλοποίησης κάθε φάσης. Επεξηγηματικά, το πόσες μονάδες θα λειτουργούν ταυτόχρονα σε κάθε φάση είναι στοιχείο εφαρμογής της τεχνικής του *unfolding*. Η τεχνική αυτή έχει ανάλογες επιδράσεις στην απόδοση του accelerator. Δυστυχώς, όμως, αντίστοιχη είναι και η επίδραση της τεχνικής αυτής στους απαιτούμενους πόρους του FPGA. Για το λόγο αυτό χρειάζεται μεγάλη προσοχή στον τρόπο και τα σημεία μίας σχεδίασης που χρησιμοποιείται η εν λόγω τεχνική.

Από την άλλη, η τεχνική του *pipeline* έχει ανάλογα αποτελέσματα στο κομμάτι της απόδο-



Σχήμα 5.13: Συνδυασμός Υπολογιστικών Φάσεων και τελική αρχιτεκτονική Hardware Accelerator



Σχήμα 5.14: Pipeline Stages εξωτερικά των υπολογιστικών φάσεων

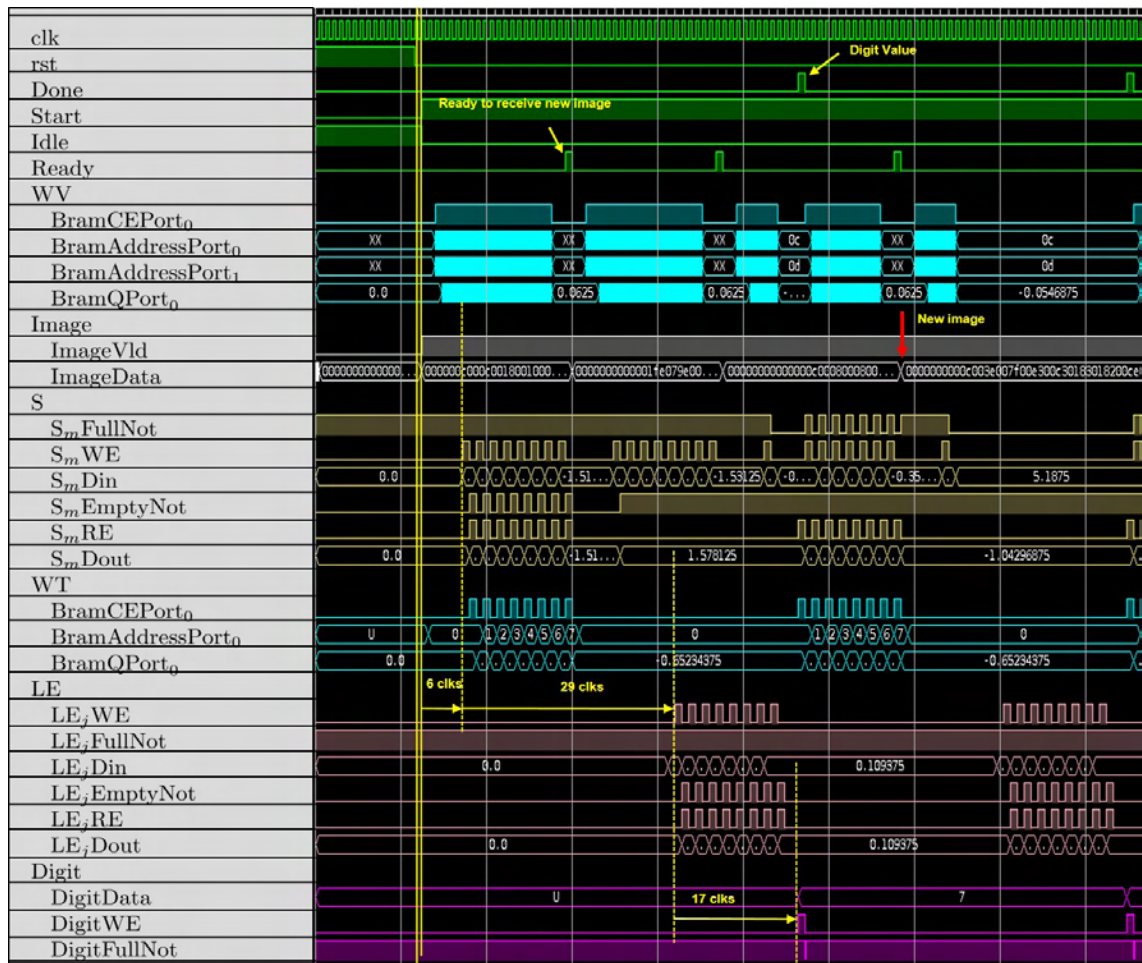
σης, με το πλεονέκτημα ότι προκαλεί μικρή επιβάρυνση στους απαιτούμενους πόρους, κυρίως σε *registers* και *LookUp Tables*. Η τεχνική αυτή μπορεί να εφαρμοστεί σε διακριτά επίπεδα κατά τη σχεδίαση και την υλοποίηση. Συγκεκριμένα για τον RBM Hardware Accelerator εφαρμόστηκε σε δύο επίπεδα, *εσωτερικά* και *εξωτερικά* των κυκλωμάτων των υπολογιστικών φάσεων. Το πρώτο επίπεδο αναλύθηκε εκτενώς και επαρκώς στην παρουσίαση της εκάστοτε φάσης. Στο σημείο αυτό θα περιγραφεί η μέθοδος εφαρμογής του pipeline στο επίπεδο του συνολικού accelerator.

Το χρονικό διάγραμμα του σχήματος 5.14 εξηγεί τη μέθοδο που εφαρμόστηκε. Αυτή ουσιαστικά διαχωρίζει το κύκλωμα σε τρία (3) στάδια, ένα για κάθε φάση. Κάθε φάση αρχίζει τους υπολογισμούς της από τη στιγμή που είναι διαθέσιμο το πρώτο αποτέλεσμα και πριν ολοκληρωθεί το σύνολο των πράξεων της προηγούμενης φάσης. Αξίζει να σημειωθεί ότι τα ενδιάμεσα αποτελέσματα αποθηκεύονται σε First-In First-Out στοιχεία, τα οποία αποτελούν και τις διεπαφές ανάμεσα στις φάσεις. Με αυτόν τον τρόπο επιτυγχάνεται συνολικό latency για τον accelerator ίσο με 53 κύκλους ρολογιού, τη στιγμή που μπορεί να μπορεί να δεχθεί

Parallel NNs	Rate in <i>MImages/s</i>
1	2.6
2	5.2
4	10.4
8	20.8

Πίνακας 5.3: *Throughput* συνολικού συστήματος με μεταβλητό αριθμό NNs

νέα εικόνα κάθε 48 κύκλους. Έτσι, με τη χρήση ρολογιού συχνότητας 125 *MHz*, ο accelerator επεξεργάζεται εικόνες με ρυθμό μεγαλύτερο από **2.5 MImages/s**. Στον πίνακα 5.3 παρουσιάζεται το συνολικό throughput που προκύπτει για ένα σύστημα με μεταβλητό αριθμό από παράλληλα νευρωνικά δίκτυα.

Σχήμα 5.15: *Accelerator Timing Waveform*

Οι κυματομορφές του σχήματος 5.15 οπτικοποιούν τα όσα αναφέρθηκαν παραπάνω και αφορούν τις βελτιστοποιήσεις όσο και τους τρόπους με τους οποίους “επικοινωνούν” οι φάσεις μεταξύ τους. Το σχήμα αυτό δίνει μία συνολική εικόνα των χρονικών στιγμών που γίνονται διαθέσιμα τα αποτελέσματα κάθε φάσης. Ακόμα, φαίνονται οι εγγραφές και οι αναγνώσεις των ενδιαμέσων αποτελεσμάτων σε *FIFOs* καθώς και οι αναγνώσεις των *weights* που είναι απαραίτητα για τους υπολογισμούς.

Primitive Resources	<i>Phase1</i>	<i>Phase2</i>	<i>Phase3</i>	<i>Accelerator</i>
<i>BRAM18K</i>	128 – 4%	0 – 0%	0 – 0%	128 – 4%
<i>DS P48E</i>	0 – 0%	440 – 12%	0 – 0%	440 – 12%
<i>FF</i>	4699 – ~ 0%	31497 – 3%	1004 – ~ 0%	37470 – 4%
<i>LUT</i>	35546 – 8%	56482 – 13%	1974 – ~ 0%	95278 – 21%

Πίνακας 5.4: *Hardware Accelerator: Resource Utilization, (Target device: Virtex 7, xc7vx690tffg1157-2)*

Τέλος, στον πίνακα 5.4 παρουσιάζονται οι απαιτήσεις σε πόρους FPGA, για κάθε φάση ξεχωριστά, καθώς και για το συνολικό HDR-RBM Hardware Accelerator.

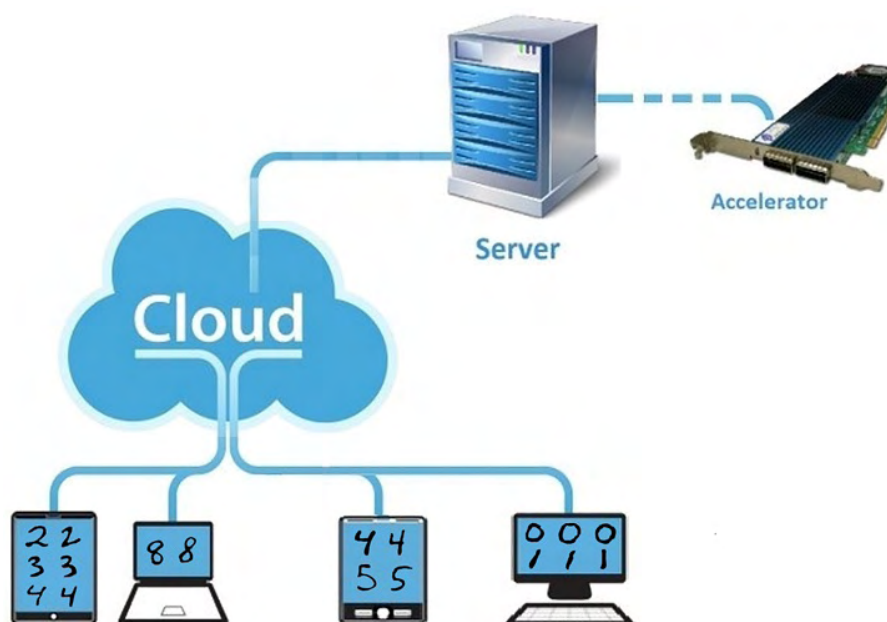
Κεφάλαιο 6

Προτυποποίηση Συστήματος και Μετρήσεις

Στο προηγούμενο κεφάλαιο 5 παρουσιάστηκε η αρχιτεκτονική του HDR-RBM Hardware Accelerator και η βιβλιοθήκη που αναπτύχθηκε με σκοπό την επιτάχυνση της διαδικασίας αυτόματης αναγνώρισης ψηφίων. Όπως εξηγήθηκε, δόθηκε ιδιαίτερη έμφαση στο σχεδιασμό του Accelerator, ώστε να εμφανίζει υψηλή προσαρμοστικότητα κατά το στάδιο της ενσωμάτωσής του σε ένα συνολικό σύστημα. Το παρόν κεφάλαιο εστιάζει σε ζητήματα που αφορούν την ολοκλήρωση του κυκλώματος του HDR-RBM Hardware Accelerator σε ένα τέτοιο σύστημα και συγκεκριμένα σε ένα *Cloud Server* ([24]). Θα εξηγηθούν επιγραμματικά οι απαιτήσεις του συνολικού συστήματος. Στη συνέχεια θα παρουσιαστούν οι πόροι που χρησιμοποιήθηκαν για την υλοποίηση, στα πλαίσια της εν λόγω διπλωματικής εργασίας. Ακόμα, θα δοθούν και θα αναλυθούν οι μέθοδοι που ακολουθήθηκαν για την επιτυχή ενσωμάτωση του Hardware Accelerator στο συνολικό σύστημα, καθώς και στοιχεία που αφορούν τους απαιτούμενους πόρους από πλευράς FPGA. Τέλος, θα παρουσιαστούν μετρήσεις σχετικά με την απόδοση του τελικού συστήματος για διαφορετικές τιμές των παραμέτρων του.

6.1 Cloud Services

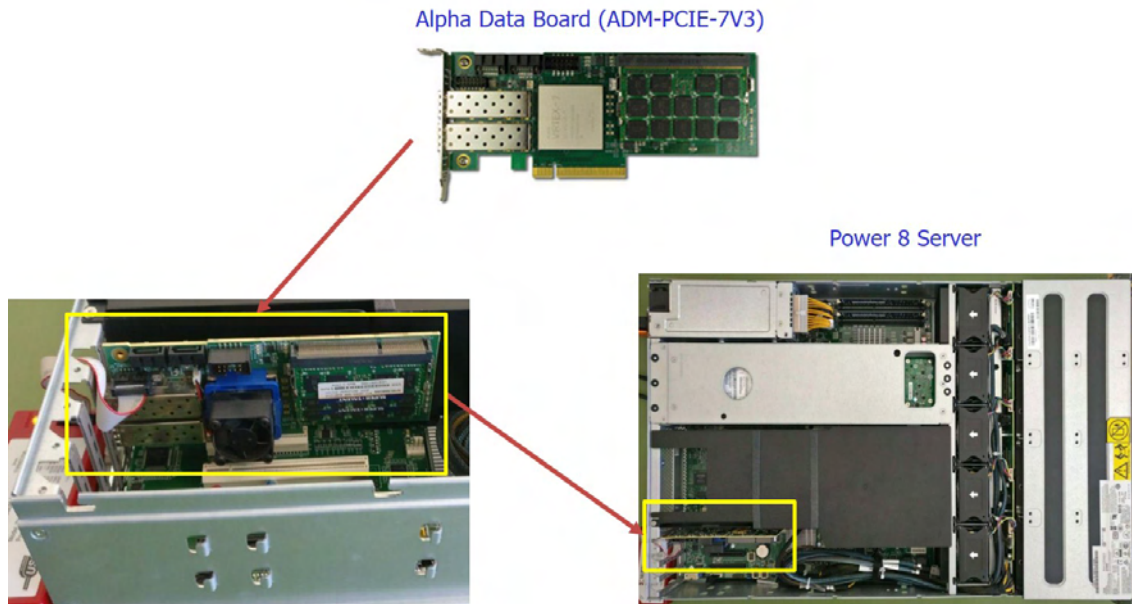
Η έννοια του *Cloud Services* καλύπτει ένα ευρύ φάσμα υπηρεσιών, που ένας πάροχος προσφέρει στους χρήστες, μέσω του διαδικτύου, γνωστό και ως *cloud* ([25], [26]). Στην περίπτωση του προβλήματος της αναγνώρισης χειρόγραφων ψηφίων, οι υπηρεσίες cloud μεταφράζονται, από την πλευρά του χρήστη, σε αποστολή εικόνων και λήψη αναγνωρισμένων ψηφίων. Οι απαιτήσεις για την κάλυψη, με επάρκεια, μίας τέτοιας υπηρεσίας, συνήθως απαγορεύουν τη χρήση ενός κλασσικού server συστήματος, όπου όλη η επεξεργασία βασίζεται στην υπολογιστική ισχύ ενός επεξεργαστή γενικού σκοπού. Μια τέτοια, λοιπόν, εφαρμογή ταιριάζει με ακρίβεια στις προδιαγραφές του Hardware Accelerator, καθώς απαιτείται υψηλός ρυθμός επεξεργασίας εικόνων και λήψης αποφάσεων. Παρακάτω θα ακολουθήσει αναλυτική περιγραφή των τεχνικών ενσωμάτωσης του Accelerator σε ένα σύστημα παροχής υπηρεσιών αυτόματης αναγνώρισης ψηφίων, μέσω cloud. Η ιδέα του RBM Cloud Service συστήματος

Σχήμα 6.1: *Cloud Services: Handwritten Digits Recognition*

περιγράφεται από την εικόνα 6.1.

6.2 Πόροι συστήματος και Εργαλεία ανάπτυξης

Αρχικά, θα παρουσιαστούν οι πόροι και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη και την υλοποίηση του τελικού συστήματος του Cloud Server, αναφορικά με το hardware. Καρδιά του συστήματος επιτάχυνσης αποτελεί το *Alpha Data PCIe Gen 3.0 (ADM-PCIE-7V3)* board, το οποίο περιλαμβάνει το *Xilinx Virtex-7 : XC7VX690T-2 - FFG1157C FPGA* chip. Το συγκεκριμένο FPGA board επιλέχθηκε, μεταξύ άλλων, διότι καλύπτει τις ανάγκες, τόσο σε διαθέσιμους πόρους (LUTs, Flip-Flops, DSP48, κλπ.), όσο και από την πλευρά των υψηλής ταχύτητας διεπαφών τις οποίες προσφέρει (PCIe - GEN3). Ένα σύστημα Cloud Services δε θα μπορούσε να ανταποκρίνεται στο όνομα αυτό, χωρίς να περιλαμβάνει τον ίδιο τον Cloud Server. Για το σκοπό αυτό αξιοποιήθηκε ο *IBM - Power8 Server* ([27]), βασικό στοιχείο του οποίου αποτελεί ο *Power8* επεξεργαστής. Ο συγκεκριμένος επεξεργαστής σχεδιάστηκε με προδιαγραφές απαιτήσεων συστημάτων εξυπηρετητών και περιλαμβάνει 8 παράλληλα Cores που λειτουργούν στα 3.32 GHz. Η τεχνολογία *SMT (Simultaneous Multithreading)* ([28]), που ενσωματώνει, επιτρέπει την ταυτόχρονη λειτουργία έως και 64 threads. Το σύστημα χρησιμοποιεί 64 GB μνήμης *DDR3* στα 1600 MHz. Η συγκρότηση του συνολικού συστήματος παρουσιάζεται στην εικόνα 6.2, όπου φαίνεται το *ADM-PCIE-7V3* board συνδεδεμένο σε μία από τις θύρες *PCIe GEN3* της *Motherboard* του *Power8 Server*. Τέλος, για την ανάπτυξη του υλικού, αξιοποιήθηκαν τα εργαλεία της *Xilinx, VIVADO Design Suite 2015.4*.

Σχήμα 6.2: *Power8 Server - Alpha Data Board*

6.3 Cloud Services: RBM Hardware Accelerator

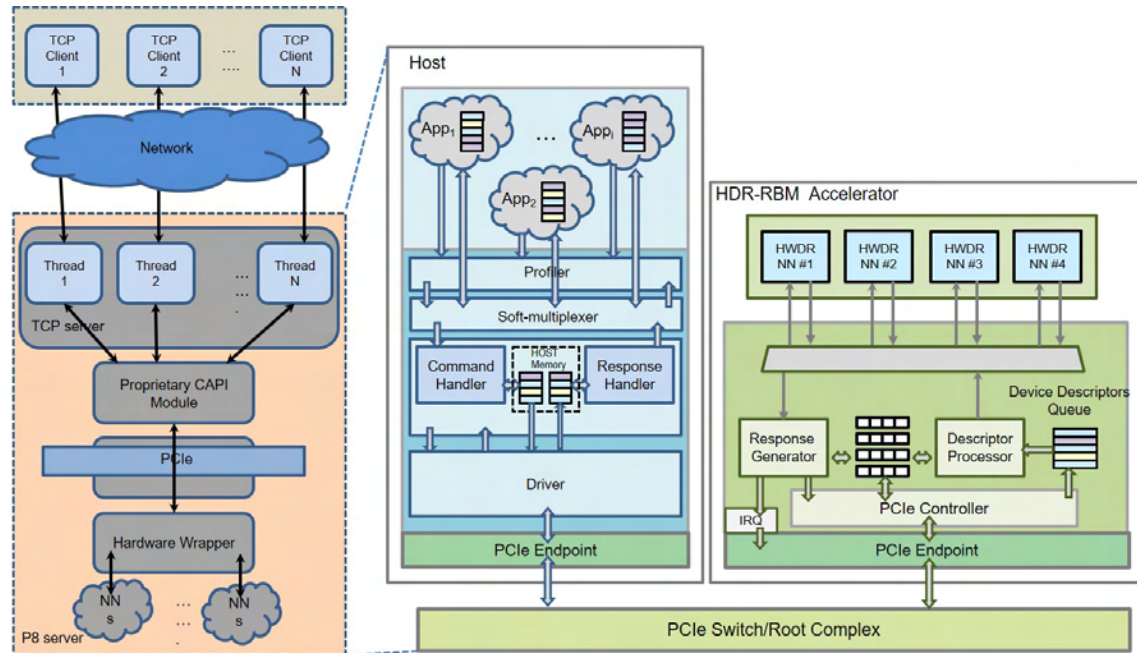
Με αφαιρετική οπτική, η υλοποίηση ενός τέτοιου συστήματος, από την πλευρά του hardware, περιλαμβάνει δύο μέρη: τον καθεαυτό Hardware Accelerator και ένα υποσύστημα που πλαισιώνει τον accelerator και του εξασφαλίζει την επικοινωνία με το γύρω περιβάλλον. Στο τελευταίο υποσύστημα υλοποιούνται όλες οι διαδικασίες των πρωτοκόλλων μεταφοράς των εικόνων από τη μνήμη του Host Server στον accelerator. Για όλες τις απαιτούμενες διαδικασίες μεταφοράς δεδομένων, επαλήθευσης και πιστοποίησης ορθής λειτουργίας του συστήματος, καθώς και λήψης μετρήσεων για τη συγκριτική μελέτη αξιοποιήθηκαν τα προϊόντα προηγούμενων εργασιών που ως κύριο αντικείμενο είχαν την υλοποίηση του λογισμικού του RBM Cloud Server [24].

6.3.1 Πρότυπα Διασύνδεσης

Στο πλαίσιο της συγκεκριμένης διπλωματικής εργασίας, η ενσωμάτωση του accelerator ακολούθησε δύο διαφορετικές αρχιτεκτονικές υλοποιήσεις, οι οποίες διαφοροποιούνται ως προς το πρωτόκολλο μεταφοράς των δεδομένων στο οποίο βασίζονται. Η πρώτη αρχιτεκτονική έκανε χρήση του κλασσικού προτύπου *PCIe - Gen3* ([29]), ενώ η δεύτερη αξιοποίησε τις δυνατότητες που προσφέρει το τελευταίας τεχνολογίας πρότυπο *CAPI (Coherent Accelerator Processor Interface)* ([30]), συμβατό με τους επεξεργαστές POWER8 της IBM. Και για τις δύο αρχιτεκτονικές είναι απαραίτητη η χρήση λογισμικού σε επίπεδο εφαρμογής λειτουργικού συστήματος, για την αποστολή και τη λήψη εικόνων και ψηφίων, αντίστοιχα, στον accelerator. Στη συγκεκριμένη εφαρμογή, πέρα από τις διαδικασίες μεταφοράς των δεδομένων, υλοποιείται η αντίστοιχη επεξεργασία με αυτή που εκτελείται στο επίπεδο του υλικού, τόσο για την ταυτοποίηση των αποτελεσμάτων, όσο και για τη σύγκριση του χρόνου εκτέλεσης των δύο υπολογισμών και τον τελικό καθορισμό της βελτίωσης στην απόδοση που επιτυγχάνεται

από την πλευρά του HDR-RBM Hardware Accelerator. Ιδιαίτερα για την περίπτωση της PCIe - Gen3 αρχιτεκτονικής, απαιτείται η χρήση εξειδικευμένου Ubuntu Kernel Driver για την επικοινωνία του επεξεργαστή με τον accelerator και την ανταλλαγή παραμέτρων και δεδομένων [31],[24].

6.3.2 Native PCIe

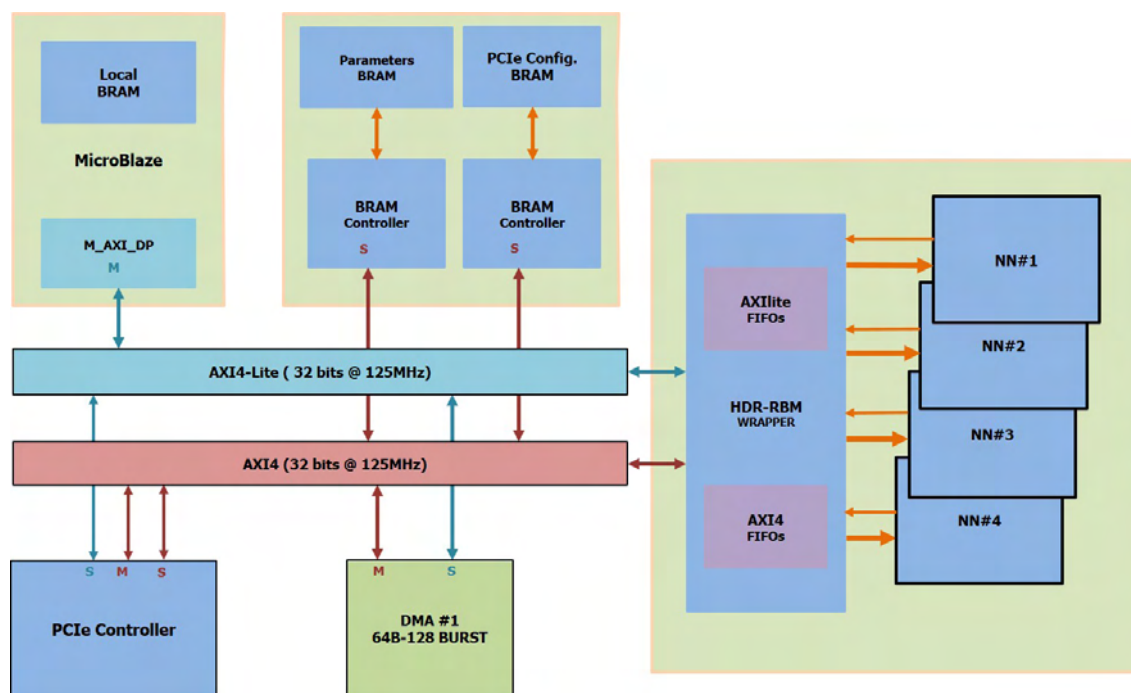


Σχήμα 6.3: Αρχιτεκτονική συστήματος Native PCIe

Η αρχιτεκτονική του συνολικού συστήματος του RBM Cloud Server over Native PCIe παρουσιάζεται στο σχήμα 6.3. Βασικό στοιχείο της αρχιτεκτονικής αυτής αποτελεί ο τρόπος επικοινωνίας του επεξεργαστή του Server με τον HDR-RBM Accelerator. Για τον σκοπό αυτό χρησιμοποιήθηκε η δομή των *Descriptors* και ο *Kernel Driver* που φαίνεται στο τμήμα του Host στο σχήμα 6.3. Ουσιαστικά για τη μεταφορά ενός *πλαίσιου* από εικόνες, που έχουν συλλεχθεί από ένα πλήθος από Clients, δημιουργείται μία δομή γνωστή ως *descriptor*. Η δομή αυτή περιλαμβάνει όλες τις απαραίτητες πληροφορίες για την ολοκληρωμένη μεταφορά των εικόνων και των ψηφίων από και προς το FPGA board. Οι ακολουθίες των ενεργειών που αφορούν τους descriptors θα παρουσιαστούν με λεπτομέρεια παρακάτω.

Είναι σημαντικό να τονιστεί ότι το (ADM-PCIE-7V3) board προσφέρει διεπαφή PCIe - Gen3, 8 lanes με μέγιστο ονομαστικό Data Rate της τάξης των 8 GBps και πραγματικό rate μεγαλύτερο των 5.5 GBps. Με αυτόν τον τρόπο υπερκαλύπτονται οι απαιτήσεις μεταφοράς των accelerator units, οι οποίες προσφέρουν ρυθμό επεξεργασίας 82.2 MBps/accelerator (2.51 MImagesps/accelerator). Η περιγραφή που θα ακολουθήσει αφορά την υλοποίηση του HDR-RBM Accelerator τμήματος του σχήματος 6.3.

Στο δομικό διάγραμμα 6.4 περιγράφεται η αρχιτεκτονική του συστήματος που υλοποιείται από τα κυκλώματα του FPGA. Αρχικά πρέπει να υπογραμμιστεί ότι η σχεδίαση που περιγράφεται στο σχήμα και περιλαμβάνει 4 νευρωνικά δίκτυα παράλληλα, είναι η τελική. Πριν



Σχήμα 6.4: Αρχιτεκτονική Native PCIe Hardware Block Design

από αυτή είχαν προηγηθεί παρόμοιες υλοποιήσεις, αλλά με 1 και 2 *RBM accelerators* να λειτουργούν ταυτόχρονα. Ουσιαστικά, η κύρια διαφοροποίηση ανάμεσα στις τρεις υλοποιήσεις αποτελεί το μέγεθος του *bus* με το οποίο τροφοδοτούνται τα νευρωνικά δίκτυα με εικόνες. Στην περίπτωση των τεσσάρων δικτύων, ο τελικός διάυλος έχει πλάτος 1024 *bits*. Όπως φαίνεται από την εικόνα 6.4, τα *NNs* πλαισιώνονται από μία δομή που ονομάζεται *HDR – RBM wrapper*, η οποία είναι υπεύθυνη για τον έλεγχο της λειτουργίας των δικτύων, την τροφοδότησή τους με εισόδους και την ανάγνωση των αποτελεσμάτων. Για τους σκοπούς αυτούς αξιοποιούνται *FIFOs* για προσωρινή αποθήκευση των δεδομένων εισόδου και εξόδου, αλλά και για εντολές ελέγχου. Συντονιστής των διαδικασιών του συστήματος του FPGA είναι ο *MicroBlaze* επεξεργαστής που παρουσιάζεται στο διάγραμμα. Ακόμα στο σχήμα φαίνονται ο *PCIe Controller*, το *DMA Engine*, καθώς και μία σειρά από τοπικές *Block – RAMs*. Συνολικά στο επίπεδο του FPGA συστήματος χρησιμοποιήθηκε ρολόι συχνότητας 125 *MHz*. Για τις εντολές ελέγχου αξιοποιήθηκε το *AXI4 – Lite Interface*, ενώ για τη μεταφορά των δεδομένων το *AXI4 Interface* [32][33].

Τα βήματα της διαδικασίας μεταφοράς των εικόνων και των ψηφίων από τη *DRAM* μνήμη του Host Server στο FPGA board και το αντίστροφο, παρουσιάζονται επιγραμματικά παρακάτω:

HOST : Αποθήκευση των εικόνων σε συγκεκριμένη διεύθυνση στην Host DRAM.

HOST : Δημιουργία descriptor, ο οποίος περιλαμβάνει τη διεύθυνση της Host DRAM όπου βρίσκονται οι εικόνες, αλλά και το πλήθος των εικόνων προς αναγνώριση. Οι descriptors που δημιουργούνται αποθηκεύονται σε συγκεκριμένη διεύθυνση στην Host DRAM, η οποία είναι αφιερωμένη στους descriptors.

HOST : Ειδοποίηση του *MicroBlaze* για τις νέες διαθέσιμες εικόνες, μέσω της εγγραφής

συγκεκριμένου πεδίου στην *PCIe Configuration BRAM* που φαίνεται στο σχήμα 6.4. Στο πεδίο αυτό αναγράφεται ο αριθμός των διαθέσιμων descriptors προς εξυπηρέτηση.

FPGA : Προγραμματισμός του DMA Engine (από τον *MicroBlaze*), για τη μεταφορά των descriptors από την Host DRAM στη μνήμη *Parameters BRAM* που υπάρχει στο δομικό διάγραμμα 6.4.

FPGA : Ανάγνωση των descriptors από τον *MicroBlaze* και εξαγωγή των πληροφοριών που περιλαμβάνουν.

FPGA : Προγραμματισμός του DMA Engine (από τον *MicroBlaze*), για τη μεταφορά των εικόνων από την Host DRAM στις *AXI4 FIFOs*, που βρίσκονται στο *Wrapper*. Για τις μεταφορές πάνω από το *PCIe* είναι υπεύθυνος ο *PCIe Controller*.

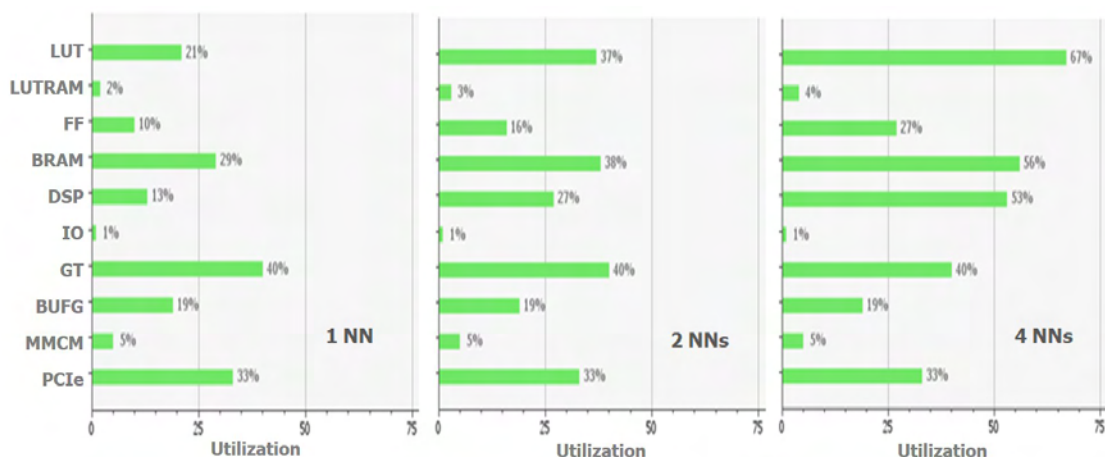
FPGA : Τροφοδότηση των RBM Accelerators με εικόνες και αναμονή των αποτελεσμάτων.

FPGA : Μετά την ολοκλήρωση των υπολογισμών, προγραμματισμός του DMA Engine για τη μεταφορά των ψηφίων από τις *AXI4 FIFOs* στην Host DRAM σε συγκεκριμένες θέσεις.

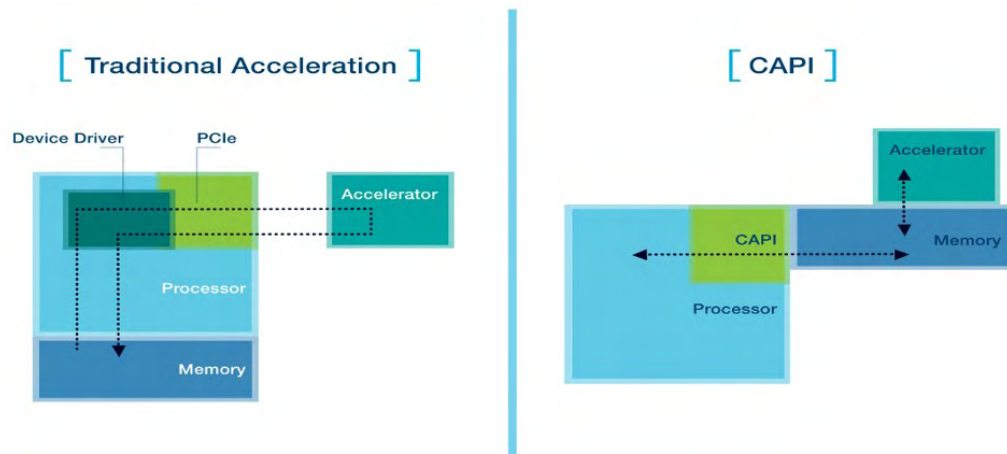
FPGA : Ανανέωση των descriptors που εξυπηρετήθηκαν και ενημέρωση του Host για τη διαθεσιμότητα νέων ψηφίων.

FPGA : Προγραμματισμός του DMA Engine (από τον *MicroBlaze*), για τη μεταφορά των ανανεωμένων descriptors από την *Parameters BRAM* στην Host DRAM στις διευθύνσεις των descriptors.

Είναι σημαντικό να τονιστεί ότι για τις διαδικασίες μεταφοράς τόσο των descriptors, όσο και των δεδομένων, επιβάλλεται η χρήση του Kernel Driver, στην πλευρά του Host. Οι διαδικασίες αυτές προσθέτουν ένα επιπλέον χρόνο στο συνολικό χρόνο εκτέλεσης. Ο πρόσθετος αυτός χρόνος ελαχιστοποιείται στην περίπτωση της υλοποίησης του *HDR-RBM Accelerator over CAPI*, όπως θα εξηγηθεί στη συνέχεια.



Σχήμα 6.5: Native PCIe FPGA Resource Utilization

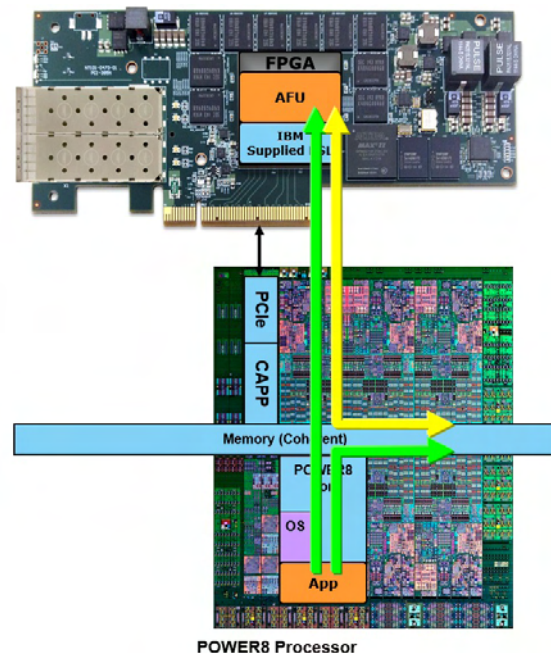


Σχήμα 6.6: CAPI Hardware Acceleration

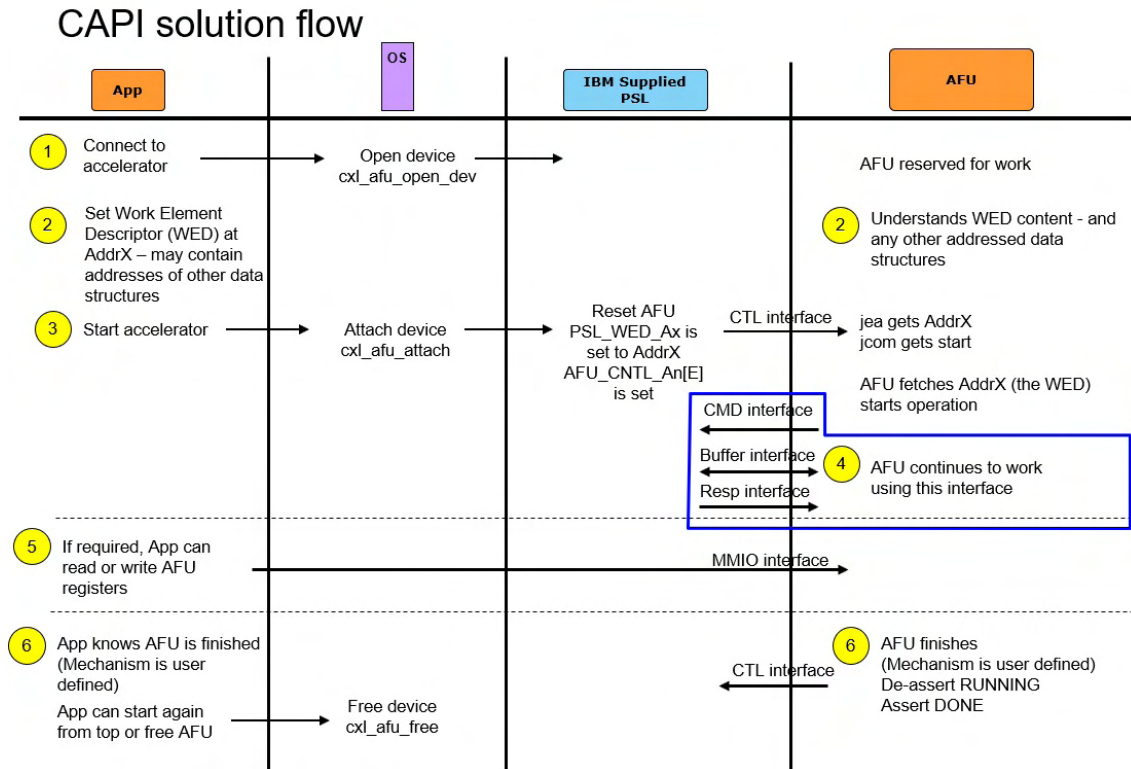
Τα διαγράμματα του σχήματος 6.5 δίνουν ορισμένες λεπτομέρειες για τους πόρους του FPGA για τις τρεις υλοποιήσεις με 1, 2 ή 4 παράλληλα NNs.

6.3.3 CAPI

Το πρότυπο διεπαφής *Coherent Accelerator Processor Interface (CAPI)* προσφέρει μία υψηλής απόδοσης λύση για την υλοποίηση εξειδικευμένων αλγορίθμων με μεγάλες απαιτήσεις υπολογιστικής ισχύος. Το CAPI καταργεί την επιβάρυνση και την πολυπλοκότητα του υποσυστήματος μεταφοράς δεδομένων, επιτρέποντας σε έναν accelerator να λειτουργεί σαν να ήταν μέρος μιας εφαρμογής σε επίπεδο λογισμικού. Μέσω της διεπαφής αυτής μειώνεται σε μεγάλο βαθμό η προγραμματιστική επιβάρυνση από την πλευρά του software με αποτέλεσμα να καθίσταται εφικτή η βελτίωση της απόδοσης ευρείας γκάμας αλγορίθμων, με τη χρήση υβριδικών συστημάτων υπολογισμού (Πολυεπεξεργαστικά συστήματα σε συνδυασμό με Custom Hardware Accelerators). Το βασικό πλεονέκτημα που προσφέρει το CAPI είναι η υλοποίηση των διαδικασιών μεταφοράς των δεδομένων πάνω από το PCIe με εξειδικευμένο υλικό που βρίσκεται πάνω στη Motherboard κάθε CAPI-enabled Server. Με αυτόν τον τρόπο είναι δυνατή η επικοινωνία της εκάστοτε εφαρμογής, που εκτελείται στον επεξεργαστή, απ' ευθείας με τον accelerator, χωρίς τη διαμεσολάβηση κάποιου Kernel Driver. Έτσι, αφαιρείται επί της ουσίας το overhead που προστίθεται από τις ενέργειες που θα έκανε ο Driver σε επίπεδο



Σχήμα 6.8: CAPI Hardware Ecosystem

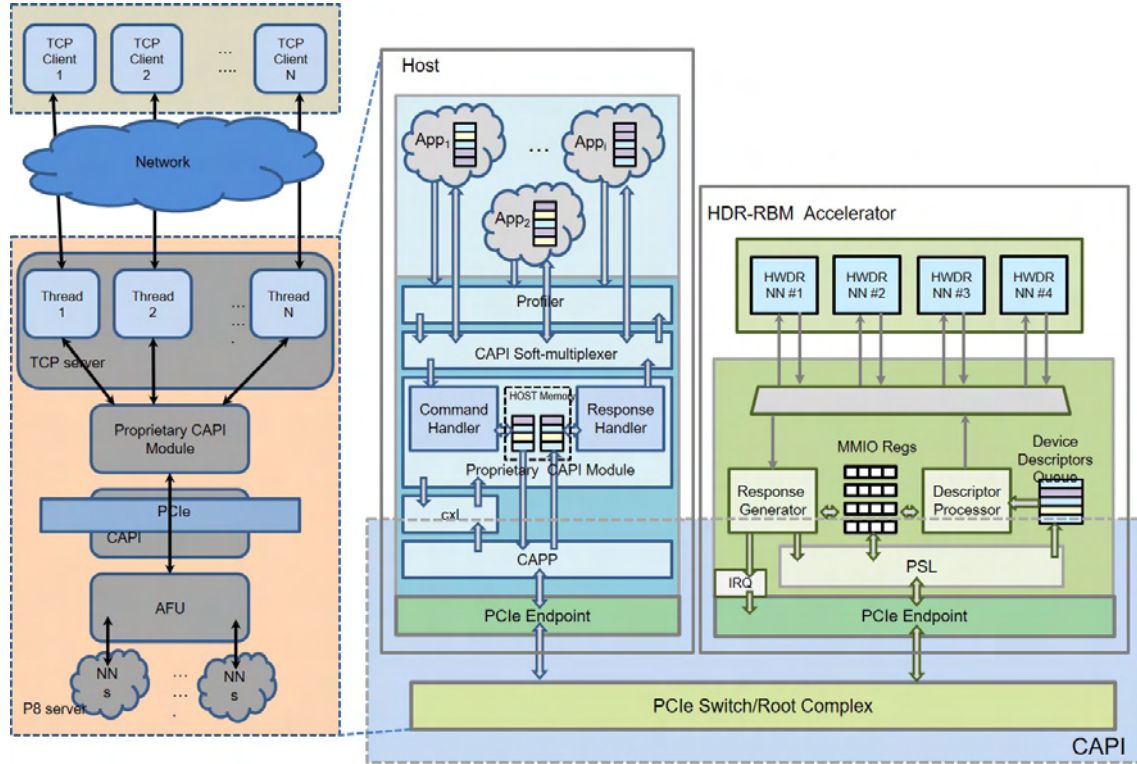


Σχήμα 6.7: CAPI Solution Flow

δο λειτουργικού συστήματος. Τα σχήματα 6.6 και 6.8 οπτικοποιούν τα πλεονεκτήματα που εμφανίζει η αρχιτεκτονική των συστημάτων που βασίζονται στο συγκεκριμένο πρότυπο.

Το σχήμα 6.9 παρουσιάζει την αρχιτεκτονική του *HDR-RBM Cloud Server over CAPI*. Από το συγκεκριμένο δομικό διάγραμμα γίνεται κατανοητό ότι, υπάρχει ένας βαθμός ομοιότητας με την αρχικό σύστημα του Native PCIe, ιδιαίτερα στα τμήματα του Software που υλοποιούν την επικοινωνία με τους Clients πάνω από το δίκτυο. Αντίστοιχα έγινε προσπάθεια να διατηρηθεί, όσο το δυνατό, η συμβατότητα των δύο αρχιτεκτονικών ως προς το κομμάτι της ενσωμάτωσης του RBM Hardware Accelerator. Για τον σκοπό αυτό, σχεδιάστηκε και για τη συγκεκριμένη υλοποίηση ένα υποσύστημα που πλαισιώνει τους παράλληλους hardware accelerators και είναι παρόμοιο με αυτό που περιγράφηκε στην προηγούμενη αρχιτεκτονική. Παρακάτω θα δουμε λεπτομέρειες που αφορούν την υλοποίηση του κυκλώματος του FPGA.

Από τα σχήματα 6.10 και 6.11 φαίνεται ότι η αρχιτεκτονική του συνολικού hardware του συστήματος έχει δομηθεί με τη λογική των επιπέδων. Πρώτο σημείο που χρήζει επεξήγησης αποτελεί το *Power Service Layer (PSL)*. Το συγκεκριμένο στοιχείο είναι απαραίτητο για την ενσωμάτωση οποιουδήποτε accelerator σε ένα CAPI-enabled σύστημα, καθώς υλοποιεί την πραγματική διεπαφή του accelerator, με το εξειδικευμένο υλικό που βρίσκεται στη Motherboard του Server, για το CAPI και ονομάζεται *Coherent Accelerator Processor Proxy (CAPP)*. Πολύ σημαντικό, ακόμα, κομμάτι της υλοποίησης του CAPI αποτελεί η *Accelerator Functional Unit*. Είναι η μονάδα που προσφέρει την επικοινωνία με το Power Service Layer (PSL) και λειτουργεί σαν μονάδα συμβατότητας ανάμεσα στον καθευατό accelerator και της διεπαφής που προσφέρεται de-facto από το PSL 6.7. Το διάγραμμα του σχήματος 6.7 περιγράφει με λεπτομέρειες τα βήματα προγραμματισμού και λειτουργίας ενός accelerator συμβατού με το



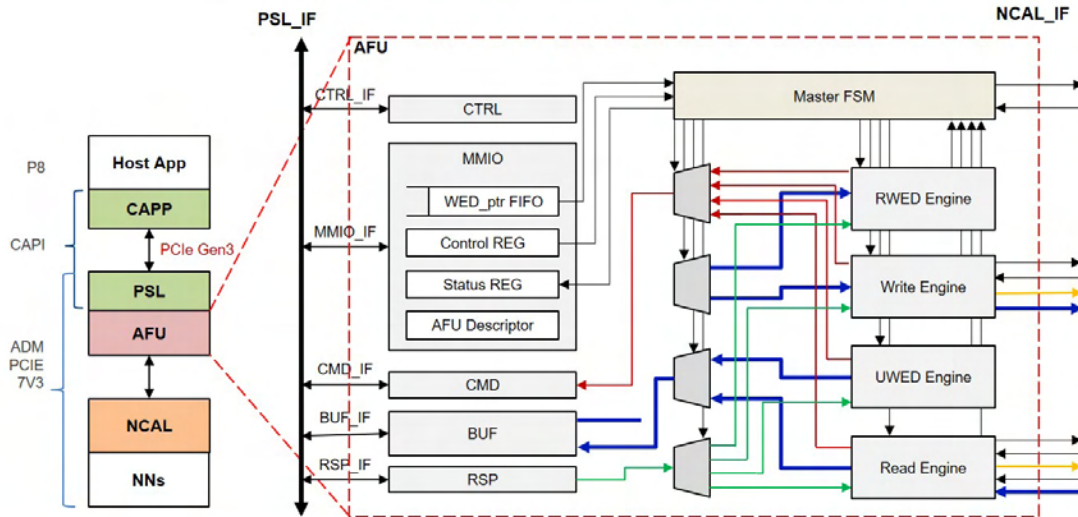
Σχήμα 6.9: Αρχιτεκτονική συστήματος CAPI

πρότυπο CAPI. Πρέπει να επισημανθεί ότι η βιβλιοθήκη, σε επίπεδο λειτουργικού συστήματος, που υλοποιεί τις απαραίτητες συναρτήσεις για τη σύνδεση και την επικοινωνία της Host CPU με τον accelerator ονομάζεται *CXL*. Ακόμα, όπως φαίνεται στο 6.7, η αντίστοιχη δομή με αυτή των descriptors, για την περίπτωση του CAPI, ονομάζεται *Work Element Descriptor (WED)*.

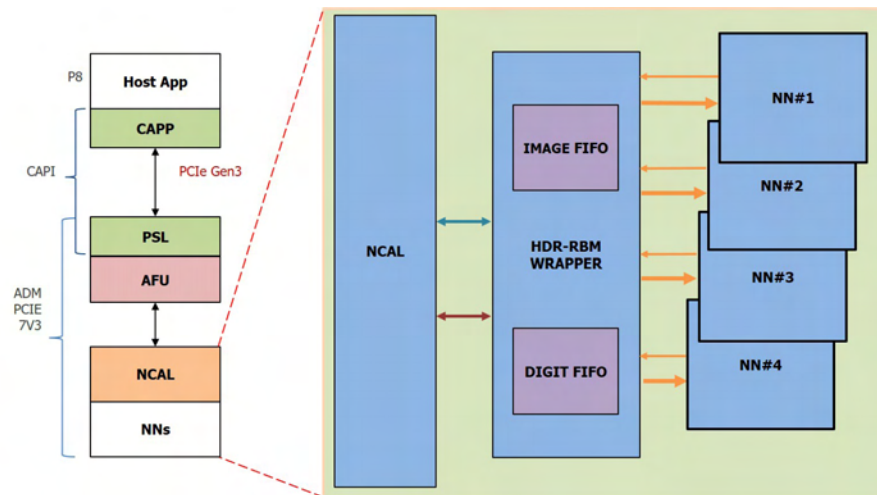
Resources	CAPI only util. %	1 NN util. %	2 NNs util. %	4 NNs util. %
LUT	17.04	30.49	42.81	69.36
LUTRAM	7.6	7.84	8.42	9.23
FF	10.44	13.43	16.7	23.46
BRAM	25.5	30.48	34.76	43.47
DSP	0	12.22	24.44	48.89
IO	8	8	8	8
GT	40	40	40	40
BUFG	34.38	34.38	34.38	34.38
MMCM	15	15	15	15
PCIe	33.33	33.33	33.33	33.33

Πίνακας 6.1: CAPI FPGA Resource Utilization

Σχετικά με την αρχιτεκτονική του FPGA συστήματος, σχεδιάστηκε μία μονάδα για την τελική ενσωμάτωση των κυκλωμάτων των νευρωνικών δικτύων που ονομάζεται *NCAL: Neural Networks CAPI Adaptation Layer*. Η μονάδα αυτή απλοποιεί τις διαδικασίες μεταφοράς των δεδομένων των εικόνων και των ψηφίων, από τους accelerators προς την AFU και το αντίστροφο. Και εδώ, είναι σημαντικό να τονιστεί ότι αναπτύχθηκαν τρεις διαφορετικές υλοποιήσεις



Σχήμα 6.10: Αρχιτεκτονική CAPI Hardware Block Design: Accelerator Functional Unit

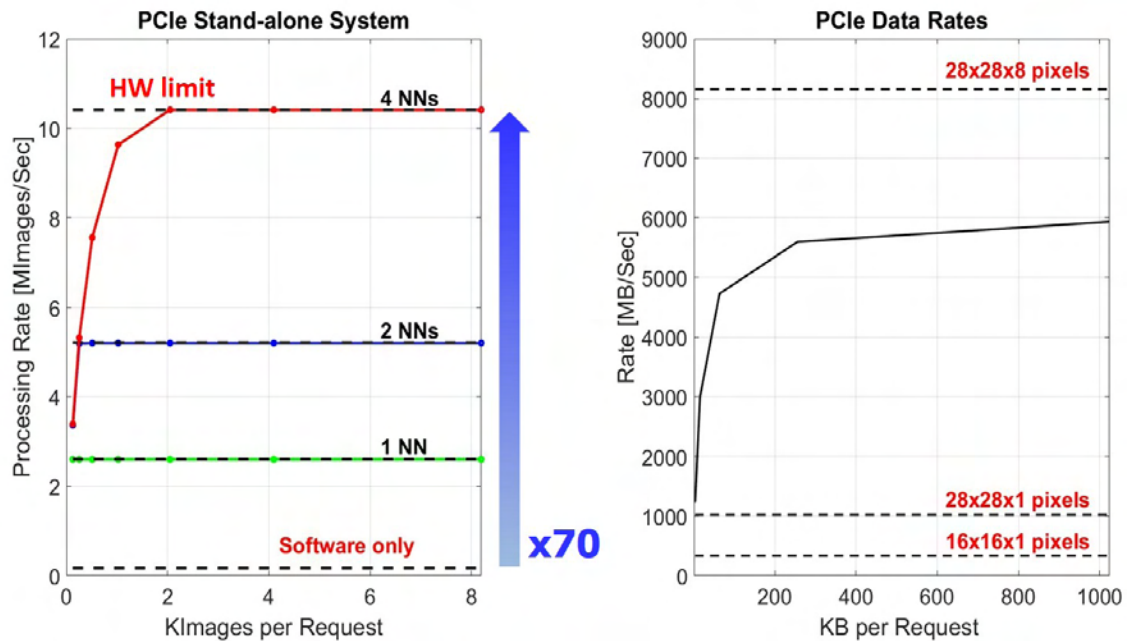


Σχήμα 6.11: Αρχιτεκτονική CAPI Hardware Block Design: Neural Networks CAPI Adaptation Layer

ενσωματώνοντας 1, 2 ή 4 HDR-RBM Hardware Accelerators, παράλληλα. Το κύκλωμα του PSL προσφέρεται από την IBM, ενώ το αντίστοιχο της AFU είναι προϊόν προγενέστερης δουλειάς [34], η οποία αξιοποιήθηκε από τον συγγραφέα του παρόντος κειμένου. Κομμάτι της υλοποίησης της συγκεκριμένης εργασίας είναι το κύκλωμα NCAL και όλα τα στοιχεία της συγκεκριμένης σχεδίασης που πλατυσώνουν και περιλαμβάνουν τους accelerators, καθώς φυσικά και ο συγκερασμός όλων των απαραίτητων στοιχείων για την επιτυχή υλοποίηση της διαδικασίας επιτάχυνσης της αυτόματης αναγνώρισης ψηφίων.

Στον πίνακα 6.1 συγκεντρώνονται τα στοιχεία που σχετίζονται με τους απαιτούμενους πόρους του FPGA για τις διαφορετικές υλοποιήσεις πάνω από το CAPI.

6.4 Συγκριτικές μετρήσεις για τις διαφορετικές υλοποιήσεις του Cloud Server: HDR-RBM Hardware Accelerator



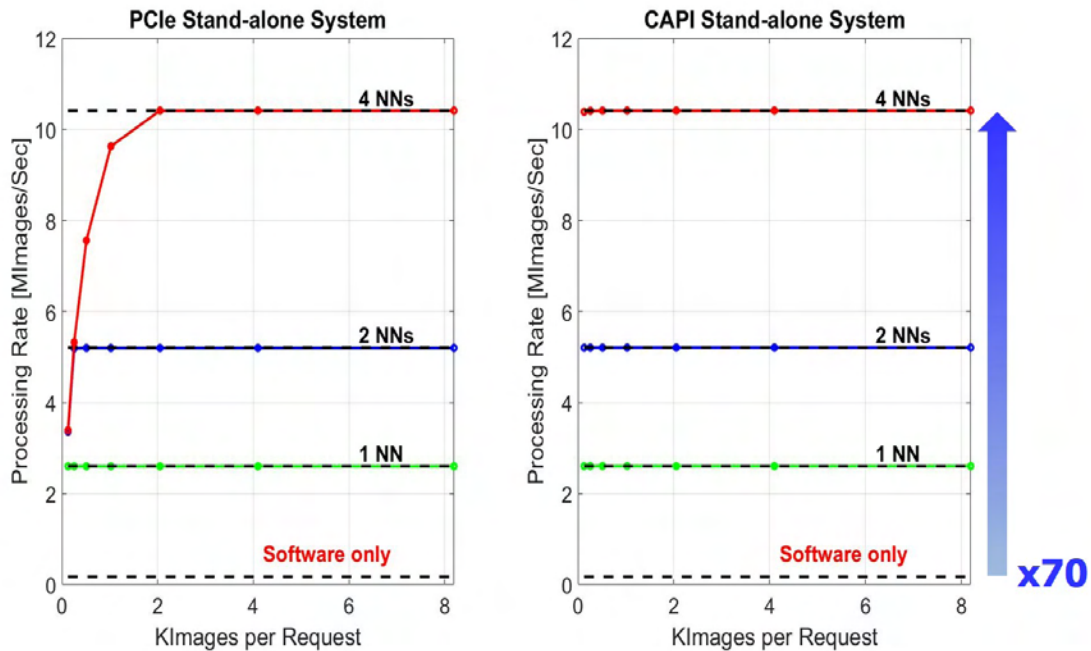
Σχήμα 6.12: Native PCIe System Acceleration

Στο σημείο αυτό θα παρουσιαστούν οι μετρήσεις που σχετίζονται με το ρυθμό επεξεργασίας που επιτυγχάνεται από το συνολικό σύστημα του Cloud Server. Οι μετρήσεις αυτές αφορούν άμεσα την εμπλοκή των RBM Hardware Accelerators και δεν περιλαμβάνουν τις μεταφορές πάνω από το δίκτυο, όπου οι εικόνες και τα ψηφία αποστέλλονται από και προς τους Clients. Ο λόγος είναι ότι το πεδίο αυτό δεν αποτελεί αντικείμενο μελέτης της παρούσας διπλωματικής εργασίας. Η απόδοση του συστήματος μετράται σε συνάρτηση του μεταβλητού πλαισίου από εικόνες που αποστέλλονται προς επεξεργασία στον accelerator. Με άλλα λόγια, μεταβάλλεται ο αριθμός των εικόνων προς επεξεργασία που περιγράφεται από τη δομή ενός descriptor, για την περίπτωση της Native PCIe υλοποίησης, ή ενός WED, για την περίπτωση της CAPI υλοποίησης.

Στο σχήμα 6.12 παρουσιάζονται δύο γραφικές παραστάσεις για την απόδοση του PCIe συστήματος. Το διάγραμμα στα δεξιά της εικόνας δίνει τα απαιτούμενα *Data Rates*, μετρούμενα σε *MB/s* για τις διαφορετικές περιπτώσεις μεγέθους εικόνας προς μεταφορά. Επιπρόσθετα, παρουσιάζονται μεγέθη εικόνων: $28 \times 28 \times 8$ (*Original MNIST*), $28 \times 28 \times 1$ (*Binarized MNIST*) και $16 \times 16 \times 1$ (*Reduced MNIST*). Είναι εμφανές ότι στην περίπτωση του *Original MNIST*, οι απαιτήσεις μεταφοράς ξεπερνούν τις δυνατότητες του PCIe - Gen3, 8 lanes.

Στην πρώτη γραφική παράσταση παρουσιάζονται συγκριτικά οι αποδόσεις, με όρους ρυθμού επεξεργασίας εικόνων, για τις περιπτώσεις της υλοποίησης αποκλειστικά με τη χρήση software, το οποίο εκτελείται σαν εφαρμογή στον POWER8 Server και τη χρήση 1, 2 ή 4 νευρωνικών δικτύων παράλληλα, σε επίπεδο hardware. Όπως χαρακτηριστικά φαίνεται από την εικόνα, η τελική υλοποίηση των 4 accelerators βελτιώνει έως και 70 φορές την εκτέλεση

των αλγορίθμων συγκρινόμενη με την υλοποίηση σε επίπεδο λογισμικού. Συγκεκριμένα, το τελικό σύστημα του RBM Hardware Accelerator προσφέρει ρυθμό επεξεργασίας της τάξης των **10.4 MImages/s**.



Σχήμα 6.13: Native PCIe and CAPI System Acceleration

Τέλος, στην εικόνα 6.13, παρουσιάζονται συγκριτικά οι αποδόσεις των δύο υλοποιήσεων του Native PCIe και του CAPI. Από τις δύο γραφικές παραστάσεις γίνεται κατανοητό το πλεονέκτημα που παρουσιάζει το CAPI λόγω του μικρότερου overhead. Επεξηγηματικά, φαίνεται ότι για Requests με μικρό αριθμό από Images, οι διαδικασίες του software, που περιλαμβάνουν τις μεταφορές πάνω από το PCIe με τη διαμεσολάβηση του Driver, λειτουργούν ως bottleneck, εμποδίζοντας το σύστημα να φτάσει στα όρια επεξεργασίας που προσφέρονται από τους hardware accelerators. Κάτι τέτοιο όμως δεν συμβαίνει στην περίπτωση του CAPI, όπου οι εν λόγω διαδικασίες αναλαμβάνονται από εξειδικευμένο υλικό στην πλευρά του Host Server. Με αυτό τον τρόπο είναι δυνατή η μέγιστη επιτάχυνση της επεξεργασίας, από το ελάχιστο μέγεθος πλαισίου εικόνων ανά Request. Τα παραπάνω αποδεικνύουν τη χρησιμότητα του πρότυπου CAPI σε εφαρμογές με υψηλή απαίτηση σε ρυθμό επεξεργασίας και υπολογιστική ισχύ, αλλά και περιβάλλοντα υβριδικής πολυ-επεξεργασίας.

Κεφάλαιο 7

Επίλογος

7.1 Συμπεράσματα

Στο πλαίσιο της διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε μία βιβλιοθήκη για τη βελτίωση του χρόνου εκτέλεσης για την αυτόματη αναγνώριση χειρόγραφων ψηφίων με τη χρήση των αλγορίθμων Restricted Boltzmann Machines και υλικού (Hardware - FPGA). Παράλληλα μελετήθηκαν μία σειρά από παραμέτρους που επηρεάζουν την απόδοση καθώς και την ακρίβεια του συστήματος, ενώ παρουσιάστηκαν όλες οι σχεδιαστικές επιλογές που οδήγησαν στην τελική έκδοση της βιβλιοθήκης HDR-RBM Hardware Accelerator. Συγκεκριμένα αναλύθηκε η επίδραση που ασκούν στην ακρίβεια του αλγορίθμου: το μέγεθος της εικόνας προς αναγνώριση, ο αριθμός των hidden nodes από τους οποίους απαρτίζεται το δίκτυο, καθώς και η αλλοίωση-τροποποίηση των δεδομένων εισόδου τόσο για τη διαδικασία του training όσο και του classification.

Σε επίπεδο σχεδίασης αναλύθηκε η επιροή της χρήσης Fixed/Floating-point αριθμητικής στην ακρίβεια και επιλέχθηκαν οι πράξεις στις οποίες θα αξιοποιηθεί το κάθε είδος αριθμητικής. Ακόμα φάνηκε ότι οι βελτιστοποιήσεις που εφαρμόστηκαν με τη χρήση του εργαλείου Vivado HLS οδήγησαν σε σημαντική αύξηση του συνολικού επεξεργαστικού ρυθμού που προσφέρει το σύστημα.

Συμπερασματικά, η διπλωματική εργασία διεκπεραίωσε τον αρχικό στόχο της βελτίωσης της απόδοσης και συγκεκριμένα: ο τελικός HDR-RBM accelerator πέτυχε αύξηση της ταχύτητας επεξεργασίας **70 φορές** σε σύγκριση με την αντίστοιχη υλοποίηση σε Software και τη χρήση επεξεργαστή γενικού σκοπού.

7.2 Μελλοντικές επεκτάσεις

Ορισμένες προτάσεις για μελλοντικές επεκτάσεις που θα μπορούσαν να εφαρμοστούν στο σύστημα του HDR-RBM accelerator παρατίθενται στη συνέχεια:

1. Υλοποίηση *Hardware Accelerator trainer*, με στόχο ο accelerator να εμπλουτιστεί με τη διαδικασία του training, η οποία μπορεί να συμβαίνει είτε στην κανονική λειτουργία του συστήματος (*online*), είτε όχι (*offline*).

2. Επέκταση του συστήματος με τη δυνατότητα αναγνώρισης γραμμάτων.
3. Τροποποίηση του accelerator για την εφαρμογή του σε SVHN dataset και The Chars74K dataset.
4. Υλοποίηση Hardware Image reducer, με στόχο ο accelerator να είναι σε θέση να δεχθεί ως εισόδους τις εικόνες του αρχικού MNIST Dataset.

Βιβλιογραφία

- [1] Rodrigo Benenson. *Classification datasets results*. https://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html.
- [2] Yann LeCun, Corinna Cortes και Christopher Burges. *MNIST handwritten digit database*. 2010.
- [3] *The Street View House Numbers (SVHN) Dataset*. <http://ufldl.stanford.edu/housenumbers/>.
- [4] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco και Andrew Y. Ng Bo Wu. *Reading Digits in Natural Images with Unsupervised Feature Learning*. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
- [5] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu και Andrew Y. Ng. *Character Recognition in Natural Images*. <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>, 2009.
- [6] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun και Rob Fergus. *Regularization of Neural Networks Using Dropconnect*. *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13*, 2013.
- [7] Alex Smola και S.V.N VishwanathanKevin. *Introduction to Machine Learning*. Cambridge University Press, Cambridge, UK, 2008.
- [8] Asja Fischer και Christian Igel. *An Introduction to Restricted Boltzmann Machines*. τόμος 7441 στο *Lecture Notes in Computer Science*, σελίδες 14–36. Springer, 2012.
- [9] Asja Fischer και Christian Igel. *Training Restricted Boltzmann Machines*. *Pattern Recogn.*, 47(1):25–39, 2014.
- [10] Hugo Larochelle και Yoshua Bengio. *Classification Using Discriminative Restricted Boltzmann Machines*. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, σελίδες 536–543, Helsinki, Finland, 2008.
- [11] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato και Fu-Jie Huang. *A Tutorial on Energy-Based Learning*. *Predicting Structured Data*G. Bakir, T. Hofman, B. Schölkopf, A. Smola και B. Taskar, επιμελητές. MIT Press, 2006.
- [12] Hugo Larochelle, Michael Mandel, Razvan Pascanu και Yoshua Bengio. *Learning Algorithms for the Classification Restricted Boltzmann Machine*. *J. Mach. Learn. Res.*, 13, 2012.

- [13] Geoffrey E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [14] Geoffrey Hinton. *Training Products of Experts by Minimizing Contrastive Divergence*. *Neural Computation*, 2000.
- [15] Sang Kyun Kim, Lawrence C. Mcabee, Peter L. McMahon και Kunle Olukotun. *A Highly Scalable Restricted Boltzmann Machine FPGA Implementation*. *Proceedings of the International Conference on Field Programmable Logic and Applications*, σελίδες 367–372, 2009.
- [16] Dan Ciresan, Ueli Meier και Jürgen Schmidhuber. *Multi-column deep neural networks for image classification*. *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, σελίδες 3642–3649, 2012.
- [17] Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella και Jürgen Schmidhuber. *Flexible, High Performance Convolutional Neural Networks for Image Classification*. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two, IJCAI'11*, σελίδες 1237–1242, 2011.
- [18] E. Bougioukou, N. Toulgaridis, M. Varsamou και T. Antonakopoulos. *Hardware Acceleration on Cloud Services: The use of Restricted Boltzmann Machines on Handwritten Digits Recognition*. *Advances in Science, Technology and Engineering Systems Journal*, 3, 2018.
- [19] N. Toulgaridis, E. Bougioukou και T. Antonakopoulos. *Architecture and Implementation of a Restricted Boltzmann Machine for Handwritten Digit Recognition*. *Proceedings of the 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, 2017.
- [20] N. Toulgaridis, E. Bougioukou και T. Antonakopoulos. *RBM-based hardware accelerator for handwritten digits recognition*. *2017 Panhellenic Conference on Electronics and Telecommunications (PACET)*, 2017.
- [21] Alpha Data Parallel Systems Ltd. *ADM-PCIE-7V3 User Manual*. <https://www.alpha-data.com/pdfs/adm-pcie-7v3%20user%20manual.pdf>.
- [22] Xilinx Inc. *Floating-Point Operator*. https://www.xilinx.com/products/intellectual-property/floating_pt.html.
- [23] Xilinx Inc. *Vivado Design Suite 7 Series FPGA Libraries Guide*, UG953. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2012_2/ug953-vivado-7series-libraries.pdf.
- [24] E. Bougioukou, N. Toulgaridis και T. Antonakopoulos. *Cloud Services using Hardware Accelerators: The case of Handwritten Digits Recognition*. *Proceedings of the 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, 2017.
- [25] Surat Teerapittayanon, Bradley McDanel και H T. Kung. *Distributed Deep Neural Networks over the Cloud, the Edge and End Devices*. 2017.

- [26] Hai jun Zhang και Nan feng Xiao. *Parallel implementation of multilayered neural networks based on Map-Reduce on cloud computing clusters*. 2015.
- [27] IBM Power Systems. *Quick Reference Guide for POWER8™ Processor-based Servers*. <https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=P0Y03114USEN>.
- [28] Alex Mericas. *Performance Characteristics of the POWER8™ Processor, IBM Systems and Technology Group Development*. https://www.hotchips.org/wp-content/uploads/hc_archives/hc26/Hc26-12-day2-epub/Hc26.12-8-Big-Iron-Servers-epub/Hc26.12.817-POWER8-Mericas-IBM%20Revised-no-spec.pdf.
- [29] PCI SIG. *PCI Express Base Specification, Revision 2.1*, 2009.
- [30] *A Deeper Look at POWER8 CAPI and Data Engine for NoSQL*, IBM Systems Magazine. <http://ibmsystemsmag.com/power/businessstrategy/competitiveadvantage/capi-deep-look/>, 2015.
- [31] E. Bougioukou και T. Antonakopoulos. *A Hybrid Device Driver for Next-Generation Solid-State Drives. The Pan-Hellenic Conference on Electronics and Telecommunications (PA-CET)*, Ioannina, Greece, 2015.
- [32] Xilinx Inc. *AXI Reference Guide*. https://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf.
- [33] ARM. *AMBA® AXI Protocol v1.0*. http://mazzola.iit.uni-miskolc.hu/~drdani/docs_arm/AMBAaxi.pdf.
- [34] A. Prodromakis, N. Papandreou, E. Bougioukou, U. Egger, N. Toulgaridis, Th. Antonakopoulos, H. Pozidis και E. Eleftheriou. *Controller Architecture for Low-latency Access to Phase-Change Memory in OpenPOWER Systems. The International Conference on Field-Programmable Logic and Applications (FPL)*, Lausanne, Switzerland, 2016.

