Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un'implementazione efficiente di una rete neurale MLP su architetture riconfigurabili FPGA

V. Conti - S.Vitabile –F. Gennaro – F. Sorbello

Rapporto Tecnico N.:14
RT-ICAR-PA-04-17 **dicembre 2004**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Implementazione di una rete neurale feed-forward su una farm di processori di tipo TotemRISC

V. Conti[2] –S.Vitabile[1] – F. Gennaro[2] – F. Sorbello[2]

Rapporto Tecnico N.:14 ***Data:***
RT-ICAR-PA-04-17 **dicembre 2004**

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sezione di Palermo Viale delle Scienze edificio 11 90128 Palermo
[2] Università degli Studi di Palermo Dipartimento di Ingegneria Informatica Viale delle Scienze 90128 Palermo

Indice

CAPITOLO 1 1
RETI NEURALI 1
1.1 INTRODUZIONE 1
1.1 CENNI STORICI. 2
1.2 TIPICO PROBLEMA DEL CLASSIFICATORE 6
1.3 TOPOLOGIA DI RETI NEURALI 7
1.4 FASE DI APPRENDIMENTO 8
1.5 RETI NEURALI SU DISPOSITIVI FPGA 11

CAPITOLO 2 13
LA STRUTTURA DEL TOTEMRISC 13
2.1 INTRODUZIONE 13
2.2 IL MICROPROCESSORE RISC 15
2.3 IL CO-PROCESSORE TOTEM 15

CAPITOLO 3 17
ANALISI DEL PROCESSORE TOTEMRISC PER L'IMPLEMENTAZIONE DI RETI NEURALI 17
3.1 INTRODUZIONE 17
3.2 CARATTERISTICHE DEL CHIP 18
3.3 ALGORITMO REACTIVE TABU SEARCH 19
3.4 AMBITI APPLICATIVI 20
3.5 VANTAGGI DI UTILIZZO 21
3.6 CENNI SULL'EVOLUZIONE DEL PRODOTTO 21
3.7 POSSIBILI SVILUPPI FUTURI 22

BIBLIOGRAFIA 24

APPENDICE A 25

Capitolo 1

Reti Neurali

1.1 Introduzione

La computazione neurale si ispira ai sistemi neurali biologici, dei quali cerca di riprodurne artificialmente la struttura e di simularne le funzioni di base. Di solito i sistemi basati su reti neurali sono analizzati come una filosofia distinta in contrapposizione ai computer digitali standard di tipo Von Neumann.

I tradizionali calcolatori sono costituiti da un processore (CPU) che accentra tutta la capacità computazionale del sistema ed esegue le operazioni in una sequenza programmata. Il concetto di algoritmo come insieme di operazioni organizzate in una sequenza opportuna (o diagramma di flusso) sta alla base di questo approccio.

La filosofia delle reti neurali dall'altra parte, ispirandosi ai sistemi biologici, considera un numero elevato di unità di calcolo(detti nodi o neuroni artificiali) con una capacità computazionale elementare, che risultano essere densamente interconnessi.

Con riferimento alla fig.1.1, dal punto di vista biologico la caratteristica principale del neurone è quella di generare un potenziale elettrico che si propaga lungo l'assone (l'output del neurone), allorché l'attività elettrica al livello del corpo del neurone supera una determinata soglia. L'input al neurone proviene da un insieme di fibre chiamate dendriti: esse sono in contatto con gli assoni di altri neuroni dai quali ricevono i potenziali elettrici. Il punto di connessione fra un assone di un neurone e il dendrite di un altro neurone è chiamato sinapsi.

La sinapsi, fra le tante, ha anche la proprietà di modulare l'impulso elettrico proveniente dall'assone. Il potenziale elettrico generato da un neurone infatti è di tipo tutto-o-nulla: se l'attività elettrica del neurone supera una certa soglia si innesca l'impulso, altrimenti no; la scarica non differisce per intensità da un neurone all'altro. Il potenziale si propaga lungo l'assone e giunge alla sinapsi con il dendrite di un altro neurone. Il potenziale post-sinaptico sul dendrite dipende dalle caratteristiche biochimiche della sinapsi. In presenza dello stesso potenziale pre-sinaptico, due sinapsi diverse generano potenziali post-sinaptici differenti. Dunque la sinapsi ha la funzione di pesare il potenziale in ingresso modulandolo. I potenziali post-sinaptici si propagano attraverso i dendriti del neurone; a livello del soma si sommano. Solo se il risultato di tale somma è superiore

ad una certa soglia il neurone innesca il potenziale che si propagherà attraverso il suo assone.

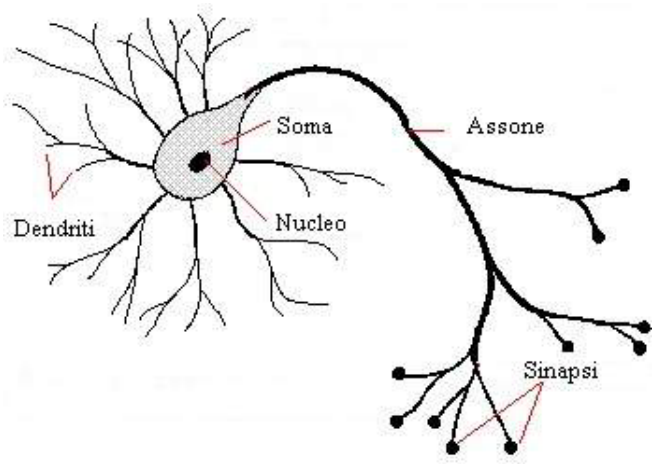

Figura 1.1: Neurone Biologico

Da un punto di vista tecnologico l'interesse per le reti neurali nasce dal fatto che esse, oltre a offrire un approccio computazionalmente favorevole a una classe di problemi che sono difficili da risolvere con metodi tradizionali, mostrano una notevole resistenza al guasto. Esse infatti continuano ad assolvere il loro compito anche in seguito a rilevante danno parziale, sia pure con prestazioni ridotte, mostrando in ciò una notevole analogia col tessuto cerebrale umano.

Le reti neurali offrono un'alternativa alla necessità di generare esplicitamente l'insieme analitico delle relazioni ingresso-uscita. Esse non richiedono che le relazioni ingresso-uscita siano esplicitamente codificate in un programma. Gli algoritmi neurali specificano una procedura di apprendimento per generare la corretta associazione tra ingressi ed uscite.

1.1 Cenni storici.

Sebbene non tutte le reti artificiali siano il modello matematico di reti neurali naturali, generalmente

con esse si desidera emulare in maniera semplificata i sofisticati processi svolti dal cervello umano, come l'analisi qualitativa di dati provenienti dall'esterno, o la capacità di apprendimento a mezzo di esempi; quest'ultimo aspetto soprattutto evidenzia la differenza tra reti neurali e calcolatori elettronici (i calcolatori necessitano sempre di una minuziosa programmazione anche per effettuare elaborazioni minime).

La prima formalizzazzione del neurone artificiale si deve a S. McCulloch e W. Pitts nel ’43, fig 1.2.

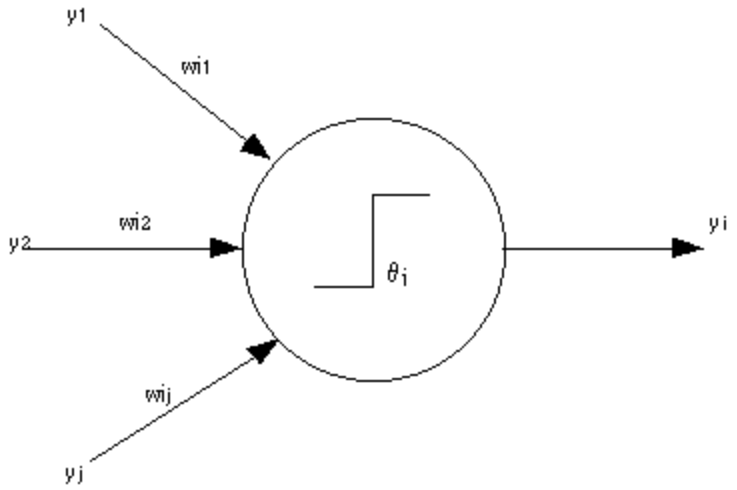

Fig. 1.2: Il modello di neurone proposto da McColluch e Pitts.

L'uscita yi del neurone è calcolata eseguendo la somma pesata dei segnali afferenti da altri neuroni. Se tale somma supera un valore di soglia θi, tipico del neurone i-esimo, l'uscita assume, dopo un certo ritardo temporale τ, il valore 1 mentre nel caso contrario l'uscita del neurone vale 0. In forma concisa risulta:

$$y_i(t+\tau) = f(\Sigma_j w_{ij} y_j(t) - \theta_i)$$

dove f è la funzione a gradino unitario e wij è il peso sinaptico che quantifica l'influenza che il neurone j ha sul neurone i.

Nel ’49 D. Hebb fornì un ulteriore grado di libertà alle reti neurali, dando la possibilità di costruire reti neurali in cui i pesi sinaptici e le soglie dei neuroni potessero essere variati al fine di adattare la risposta della rete a variazioni dei segnali di ingresso. A differenza dei calcolatori tradizionali le reti neurali sono, in linea di principio, capaci di adattarsi a condizioni non previste nelle istruzioni del programma.
L'idea su cui si basa la regola di Hebb per la variazione del peso della connessione tra due neuroni è che se le unità i e j sono simultaneamente attivate allora il peso della loro connessione deve essere rafforzato, secondo la formula:

$$\Delta w_{ij} = \lambda\, a_i a_j$$

dove λ è una costante che rappresenta il coefficiente di apprendimento ed ai aj sono il valore delle uscite dei due neuroni.

Il lavoro di Minsky e Papert provocò una perdita di interesse nei riguardi delle reti neurali, da parte degli ambienti scientifici e industriali, che durò fino ai primi anni '80 quando, in seguito a risultati teorici importanti e allo sviluppo della microelettronica, rinasce l'interesse per le reti neurali.

Nel 1959 da Rosenblatt definì il primo modello di Percettrone. Si trattava di una rete neurale non retroazionata composta da neuroni a soglia organizzati in tre strati e con una unica uscita binaria.

Il Percettrone nella sua prima versione, nota come Percettrone semplice, è composto da m neuroni per lo strato di ingresso che alimentano uno strato composto da n elementi confluenti in un unica unità di uscita. Tale rete, effettuando una trasformazione da uno spazio ad m dimensioni ad uno unidimensionale, stabilisce a quale classe, delle due possibili, appartiene un generico ingresso che viene presentato alla rete.

La figura 1.3 riporta il Percettrone semplice proposto da Rosenblatt.

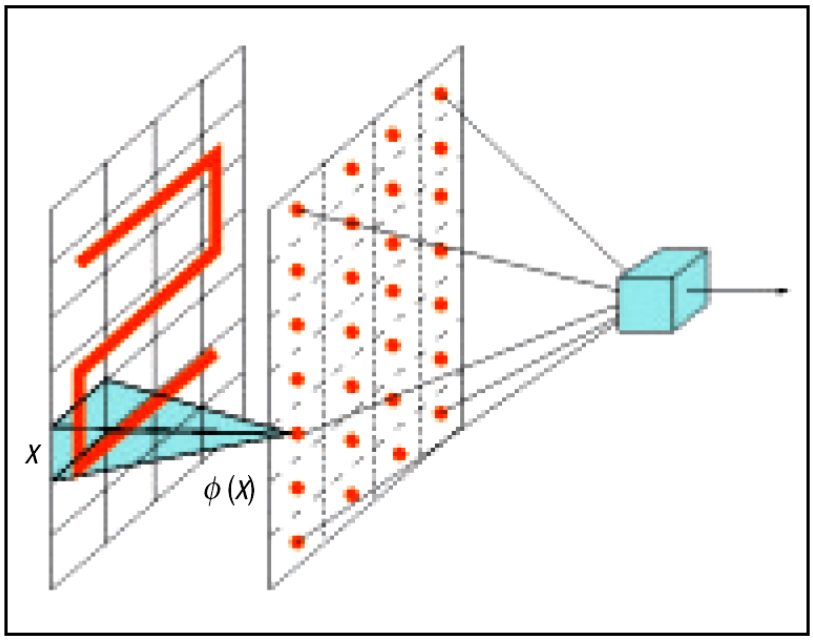

Figura 1.3: Il Percettrone semplice.

Gli ingressi da riconoscere sono presentati al primo strato del Percettrone tramite in una retina di unità sensoriali (ad es. fotocellule) sensibili alla luce: ogni unità sensoriale è attivata solo se la l'intensità luminosa rilevata supera una certa soglia. Le unità sensoriali

sono connesse ad n unità associative le cui attivazioni sono una generica funzione dei soli ingressi.
L'unità di uscita (unità decisionale) riceve in ingresso tutte le uscite delle unità associative e calcola l'uscita secondo la formula:

$$O^p = F\left(\sum_i \; w_i \Phi_i + \theta\right)$$

dove p è l'esempio presentato, □i è il valore dell'ingresso della i-esima unità associativa, wi è il peso della connessione tra l'unità di uscita e la i-esima unità associativa, θ è il valore della soglia dell'unità di uscita ed F è la funzione di Heaviside così definita:

$$F(i) = \begin{cases} 1 \text{ se } i > 0 \\ -1 \text{ se altrimenti} \end{cases}$$

Rosenblatt dimostrò che se esiste un iperpiano, nello spazio di ingresso ad m dimensioni, che può dividere i dati di ingresso in due classi distinte e separate, allora la procedura di addestramento converge in tempo finito alla soluzione; viceversa, se tale iperpiano non esiste, la procedura non converge determinando un comportamento oscillante della rete.
Appare chiaro quindi che il Percettrone semplice assicura la convergenza solo nei problemi in cui i dati di ingresso sono linearmente separabili.
Nel 1969 M. Minsky e S. Papert pubblicarono, nel loro lavoro 'Perceptrons: An Introduction to Computational Geometry', i risultati della loro approfondita analisi sul Percettrone semplice che mettevano in risalto i limiti che tale rete neurale presentava.
In tale lavoro essi dimostrarono che il Percettrone semplice non può risolvere il problema dell' or-esclusivo: infatti, anche nel caso di due soli ingressi, non si riesce a trovare una retta che separi in due classi distinte lo spazio (bidimensionale) degli ingressi.
Si può dimostrare che l'aggiunta di una o più unità nascoste alla struttura del Percettrone semplice permette la soluzione del problema dell'or-esclusivo ed in generale dei problemi non linearmente separabili. Tale rete neurale a più strati prende il nome di Percettrone Multistrato.

1.2 Tipico problema del classificatore

Lo scopo della classificazione è l'assegnazione dei pattern d'ingresso ad una o più classi di appartenenza.

A titolo di esempio, si consideri una distribuzione di pattern come quella di Figura 1.6a a palline arancione e nere, si vuole trovare due gruppi differenti in base al colore delle palline.

Come si vede, non è possibile trovare un unico piano che separi in due gruppi i pattern con target diversi, pertanto nessuna rete neurale costituita da un solo neurone, sarebbe in grado di risolvere il problema. Si rivela quindi necessario lo strato dei neuroni di hidden, il cui scopo è quello di trasformare lo spazio degli ingressi in un altro nel quale i pattern godano della separabilità, cioè di attuare divisioni preliminari dello spazio per facilitare il compito ai neuroni d'uscita.

Con questa tecnica da un insieme di partenza linearmente non separabile è possibile ottenere, dopo la pre-elaborazione ottenuta tramite gli strati nascosti, un insieme linearmente separabile.

Nel caso considerato, bastano due neuroni di hidden per attuare la separazione preliminare dei pattern, mediante i rispettivi piani h1 e h2 come mostrato in Figura 1.4a-b.

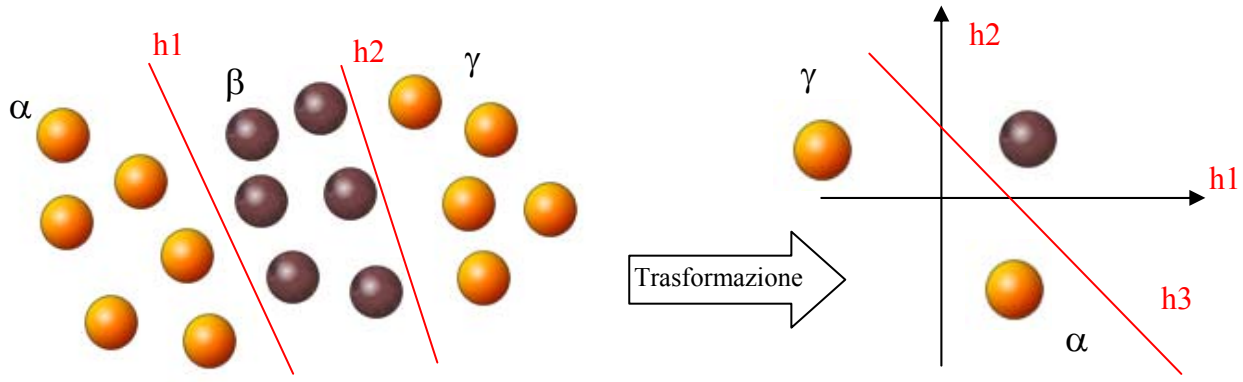

Figura 1. 4: Esempio di elaborazione dello strato di hidden

A questo punto, il neurone d'uscita riceverà in ingresso le attivazioni dei due neuroni di hidden, che saranno entrambe positive solo nella parte di spazio identificata da β .Dunque il neurone di output riceve in ingresso un pattern linearmente separabile con un unico piano , chiamato h3.

1.3 Topologia di reti neurali

Purtroppo non è possibile classificare in maniera rigorosa ed esauriente tutti i tipi di rete neurale esistenti allo stato attuale, in quanto il settore è in continua evoluzione, tanto da poter affermare che mensilmente vengono create nuove reti neurali con caratteristiche innovative rispetto alle precedenti.

Vengono riportati alcuni concetti basilari che caratterizzano le reti neurali.

Una prima caratteristica che contraddistingue una rete neurale è la topologia, ovvero la sua

struttura fisica.

In generale, una rete neurale è costituita da tante piccole unità elementari, i neuroni appunto, distribuiti in diversi strati (layers): in particolare, i neuroni intermedi costituiscono lo strato nascosto(hidden layer), i neuroni di uscita costituiscono lo strato i Output, mentre gli ingressi applicati alla rete formano lo strato di Input (Fig. 1.5).

Ciascun neurone è collegato agli altri neuroni, appartenenti allo stesso strato o a strati diversi, tramite connessioni, e ad ogni connessione è associato un valore, detto peso.

A seconda della modalità con cui questi collegamenti avvengono, possiamo dividere le reti neurali in “feed-forward” oppure “feed-back” [16].

Le reti feed-forward prevedono collegamenti monodirezionali tra neuroni appartenenti a strati

immediatamente consecutivi (connessione in avanti, cioè dallo strato n allo strato n+1): quindi gli ingressi sono collegati solamente ai neuroni dello strato intermedio (ve ne possono essere anche più di uno), i quali a loro volta sono uniti ai neuroni dello strato di uscita (Fig. 1.5).

Nelle reti feed-back, invece, sono possibili anche collegamenti tra neuroni appartenenti allo stesso strato (connessione laterale o competitiva) e collegamenti tra un neurone e i neuroni situati nello strato precedente (connessione retroattiva).

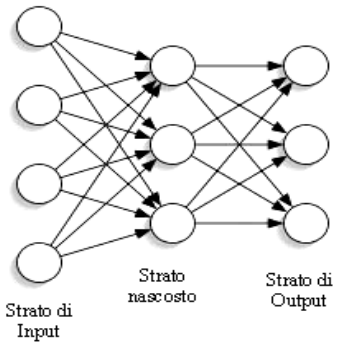

Figura 1. 5 Struttura stratificata feed-forward

Altra caratteristica fondamentale della rete neurale è la funzione di trasferimento.
Nel modello distribuito di una rete neurale l'uscita di ogni neurone è uguale alla sua attivazione, occorre quindi una funzione che, partendo dall'ingresso e dall'attivazione del generico neurone i all'istante t, produca la nuova attivazione all'istante successivo.
Le funzioni di attivazione sono funzioni non decrescenti e non lineari, le più comuni sono riportate nella successiva figura 1.3.

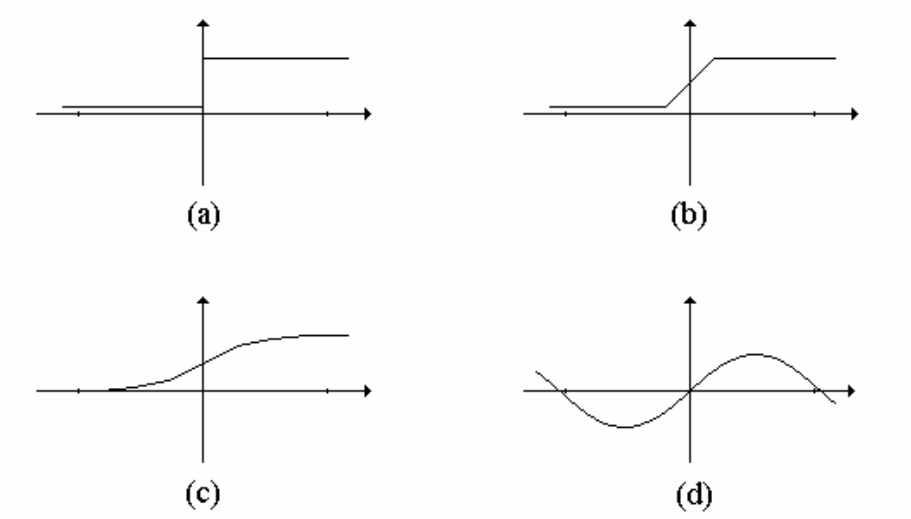

Figura. 1.6: Le principali funzioni di attivazione: (a) funzione a gradino, (b) funzione semi-lineare, (c) funzione sigmoidale, (d) funzione sinusoidale.

1.4 Fase di apprendimento

La fase di apprendimento o fase di addestramento di una rete neurale precede la fase di riconoscimento in cui la rete neurale, in base alla configurazione dei pesi raggiunta alla fine della fase di apprendimento, elabora gli ingessi che probabilmente non sono contenuti negli esempi di addestramento.
La conoscenza della rete è distribuita, infatti l'informazione acquisita dalla rete viene immagazzinata nei valori dei pesi, che possono essere considerati, nella loro globalità, come la memoria del sistema, ovvero il livello di conoscenza raggiunto.
Detto questo, risulta chiaro come l'efficienza e la precisione della rete neurale risiedano nella sua capacità di acquisire conoscenza, e ciò dipende fortemente dalla fase di addestramento (training) è proprio durante il training che i pesi vengono modificati dinamicamente, in modo da minimizzare gli errori di valutazione di un insieme prefissato di esempi.

Per essere più precisi, in base all' algoritmo di apprendimento adottato, le reti neurali vengono
distinte in reti "con supervisore" e reti "senza supervisore" [17].
L'addestramento con supervised consiste nel presentare alla rete alcuni campioni dello spazio degli ingressi (pattern), il cui insieme costituisce il training set, congiuntamente al campione dell'uscita (che rappresenta il risultato desiderato), chiamato target; la rete può così adattare i propri pesi, nel tentativo di far corrispondere le sue uscite con le uscite desiderate. Nella fase di test si possono utilizzare sequenze di dati di ingresso non presenti nell'insieme di apprendimento per osservare se la rete riesce a generalizzare correttamente quello che ha imparato.
L'idea che sta base di questo tipo di algoritmi di apprendimento, consiste nell'utilizzare una misura di errore (distanza) tra la risposta fornita dalla rete e la risposta esatta per correggere i pesi sinaptci.
I valori dei pesi sinaptici vengono quindi modificati in proporzione all'errore delle unità di output.
All'inizio dell'apprendimento i pesi sinaptici assumono piccoli valori casuali che generano quindi un output casuale per ciascun pattern di input del training set. Ogni volta che un pattern di input viene presentato, le unità di output producono una risposta che viene confrontata con la risposta desiderata; l'errore viene quindi utilizzato per modificare i pesi sinaptici.
Tutti gli algoritmi di modifica dei pesi proposti, possono essere considerati una variazione della regola di Hebb di cui ci siamo già occupati.
Regola di Widrow-Hoff. Una regola di apprendimento usata è la regola di Widrow-Hoff o regola delta in cui la variazione dei pesi delle connessioni di una rete neurale viene calcolata in modo da minimizzare la funzione di errore cosi' definita:

$$E = \sum_p E^p = 0{,}5 \times \sum_p (a^p - d^p)^2$$

dove Ep indica l'errore commesso dalla rete sull'esempio p, ed ap e dp indicano rispettivamente il valore reale ed il valore desiderato dell'uscita della rete quando viene presentato l'esempio p.
Gradiente decrescente. L'idea di base nella minimizzazione della funzione di errore tramite il metodo del gradiente decrescente consiste nel far variare il valore del peso di una connessione proporzionalmente al valore, invertito di segno, della derivata della funzione

di errore rispetto al peso stesso, ciò equivale all'individuazione della direzione di massima pendenza nello spazio dei pesi.
Tutti i pattern del training set vengono presentati più volte fino a quando la rete produce la risposta corretta o riduce l'errore medio al di sotto di una certa soglia.
L'addestramento senza supervisore è caratterizzato dal fatto che la rete non dispone del target, e quindi cerca di evincere alcune proprietà degli ingressi solo in base alla loro disposizione, realizzando in tal senso una specie di compressione dati. Nelle reti di tipo non supervisionato, si lascia il compito alla rete stessa di ricercare una classificazione logica dei dati di ingresso.
Nelle reti addestrate con regole competitive vi è una sola unità di output attiva per volta.
Ciascuna unità di output appartiene ad un insieme di unità in cui compete con le altre al fine di essere l'unica attiva. Questo tipo di organizzazione dei nodi di uscita è chiamato Winner-Take-All (il vincitore prende tutto).
Lo scopo di queste reti è di raggruppare o classificare i pattern di input in modo tale che ciascuna unità di output rappresenti una certa categoria.
Nell'apprendimento competitivo semplice vi è uno strato di unità di input e uno di output. Per ogni pattern di input ciascuna unità di output calcola la propria attivazione e solamente i pesi sinaptici afferenti all'unità vincente (quella con la massima attivazione) vengono modificati in modo da spingere il vettore dei pesi nella direzione del vettore del pattern di input.
Un altro tipo di distinzione a livello di addestramento avviene a seconda delle modalità con cui viene aggiornato il vettore dei pesi: si può parlare quindi di addestramento incrementale (by pattern) o di addestramento a lotti (o batch). Nel primo caso il vettore è aggiornato
ogni volta che viene commesso un errore, ovvero ad ogni pattern (in caso di corretta classificazione il vettore viene modificato tramite un vettore nullo); nel secondo caso il vettore viene aggiornato solo alla fine di ogni ciclo sull'intero training\ set, in base a tutti gli errori commessi.
Quest'ultimo tipo di addestramento garantisce una maggiore robustezza nei confronti di un eventuale rumore nel training set.

1.5 Reti neurali su dispositivi FPGA

Dopo avere fatto una breve introduzione sulle reti neurali e sul perché in molti campi risultano essere convenienti sia come tipo di approccio per la risoluzione dei problemi che come affidabilità, adesso ci occupiamo di come rendere tali reti efficienti.

Nonostante il numero di architetture di reti neurali artificiali è elevato, un elemento che contraddistingue tutte le reti neurali è la presenza di numerosi processi che possono essere eseguiti in parallelo.

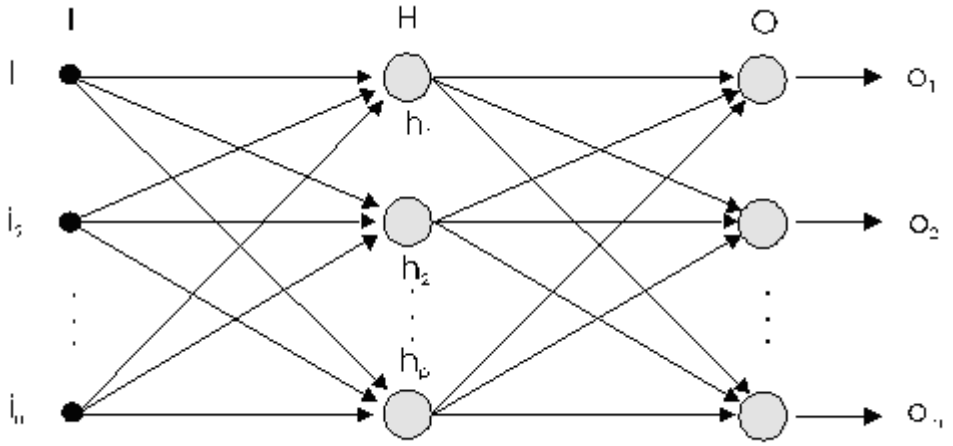

Figura 1.7: Rete multistrato

Con riferimento alla fig. 1.7 considerando una rete a tre strati, si osserva che tutti i nodi di H possono processare parallelamente gli ingressi I che giungono sequenzialmente alla rete. Inoltre come mostrato in fig. 1.8 mentre i nodi H calcolano le nuove uscite, in pipeline i nodi O possono elaborare le uscite H provenienti da un'elaborazione precedente. Dunque nel caso di ingressi consecutivi il tempo necessario per elaborare il primo ingresso è pari a T= ts1+ ts2 dove ts1 è il tempo di elaborazione del primo strato e ts2 è il tempo di elaborazione del secondo, mentre per elaborare gli ingressi successivi il tempo di elaborazione è pari al T= max(ts1, ts2) questo perché i due strati possono lavorare contemporaneamente.

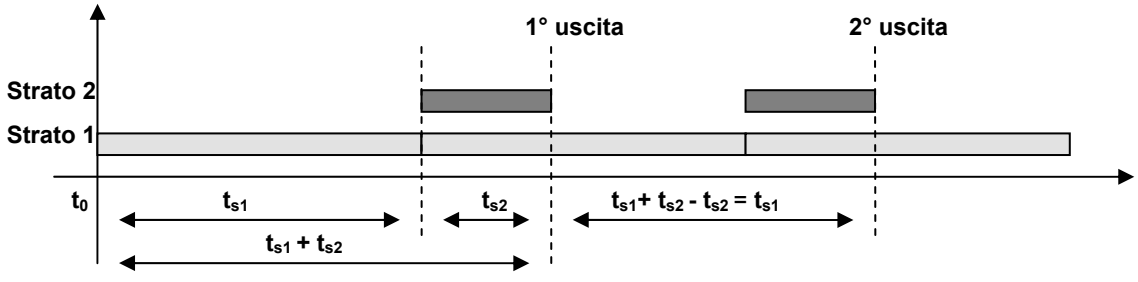

Figura 1.8: Tempi di esecuzione del 1° e 2° strato

Inizialmente per applicare le teorie sulle rete neurali si utilizzavano i calcolatori più diffusi cioè quelli basati sul la macchina di Von Neumann, tramite software che simulavano la possibilità di avere processi paralleli, ma in realtà è noto che in una macchina monoprocessore è possibile eseguire una sola istruzione per volta o al più 2 o 4 istruzioni nelle CPU più recenti ottimizzando il codice. Dunque non si sfruttava il pregio di avere tanti processi indipendenti che potevano essere eseguiti in parallelo, incrementando notevolmente l'efficienza.

Le prime sperimentazioni per realizzare una rete su hardware sono state condotte tramite sistemi nati per il calcolo distribuito, dotati di numerosi processori connessi tramite una configurazione dinamica, a secondo dell'applicazione era possibile stabilire le relazioni tra i vari processori, tipicamente un processore si comporta come uno o più neuroni connessi alla rete. Tali sistemi vengono chiamati anche Neuro-Computers [18].

Capitolo 2

La Struttura del TotemRisc

2.1 Introduzione

TotemRISC è un dispositivo di architetture distribuite di calcolatori basato sul principio che l'elaborazione dei dati digitale è meno costosa della relativa trasmissione dei dati stessi e quindi qualunque tipo di informazione dovrebbe essere convertita in formato digitale e dovrebbe essere trattata digitalmente dal relativo sensore. Questo tipo di trattamento e di approccio contribuisce anche all'integrità del segnale e ad una riduzione del rumore.
Il sistema lega sul chip un'unità neurale co-processore Totem ad un processore RISC centrale di tipo Von Neuman, ai quali a turno sono collegati un numero di analoghi sensori al fine di creare rapidamente un sistema di rete gerarchico di sensori intelligenti con nodi ciascuno dei quali capace di elaborare informazioni e propagare il relativo risultato al prossimo livello gerarchico e finire possibilmente al processore centrale.
I vantaggi di questo paradigma sono sicuramente legati alla realizzabilità e affidabilità dell'architettura. L'unità centrale non impedisce sicuramente l'esecuzione di ulteriori sotto-sistemi e ogni sensore intelligente può essere fatto lavorare in mono-tasking invece di elaborazioni parallele che notoriamente danno minori garanzie per ciò che riguarda affidabilità e realizzabilità.
Considerando che i costi extra per i processori locali stanno diventando sempre minori, mentre i moderni protocolli di comunicazione diventano sempre più complessi spesso richiedendo a loro volta processori di controllo di elevata complessità, si capisce come l'approccio discusso prima diventa più vantaggioso. L'elaborazione distribuita dei dati tramite sensori intelligenti è sicuramente preferita alla trasmissione dei dati ad una unità centrale.
In figura 1 è presentato un piccolo esempio relativo ad un modulo per elaborazione intelligente di dati di telecamera.

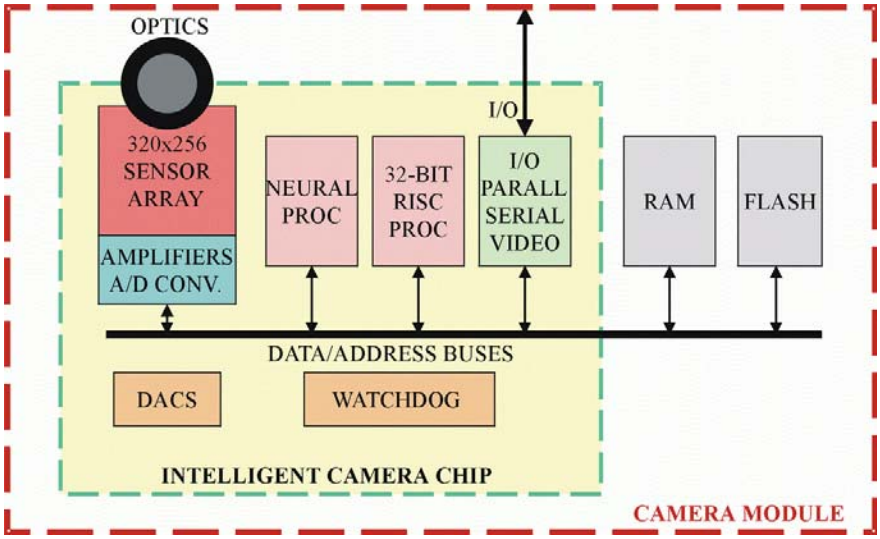

Figura 2.1: Architettura di una Macchina fotografica con sensore intelligente integrato.

L'architettura mostrata in Figura 1 è stata realizzata per introdurre maggior semplicità e programmabilità per gli scopi desiderati. Tale architettura comprende un sensore ottico digitale con possibilità di selezione di un numero di pixel e circuiti di elaborazioni di segnali, un processore RISC, un co-processore a rete neurale, logica per la gestione di memorie e relative interfacce, come sarà descritto successivamente con maggiori dettagli.
Le principali memorie di programmi e dati sono implementate all'esterno del chip, mentre tutti i componenti sono connessi tramite un singolo bus dati per maggior flessibilità. Inoltre il chip ha la caratteristica di avere un consumo di potenza.
In Figura 2 è possibile vedere un chip realizzato con le caratteristiche prima esposte, relativamente alla macchina fotografica intelligente prima descritta. L'ambiente di sviluppo del software supporta sia il C che l'Assembler.

Figura 2.2: Struttura del chip: memorie, connettori e tutti i componenti passivi collegati ad essi.

2.2 Il microprocessore RISC

Il processore principale è una macchina RISC a 32 bit dell'ARC con un insieme di istruzioni di base migliorato per eseguire direttamente delle istruzioni relative alla visione ed istruzioni di 0.5KB per la cache usate per aumentare la velocità d'esecuzione. Inoltre è anche provvisto di un totale supporto per il debug per il controllo del processore durante lo sviluppo. Il microprocessore è sintetizzato tramite una libreria di celle standard e blocchi di memoria di 32 registri a 32bit. Il processore principale controlla tutte le funzioni principali della macchina fotografica, trasferimenti in memoria e controllo periferico. La potenza di elaborazione alla velocità nominale del clock di 60 MHz è di 60 MIPS. Il processore principale lavora tramite memoria esterna dato le limitazioni dello spazio sul chip, questo offre vantaggi in termini di configurabilità della dimensione della memoria e quindi il costo del sistema. La logica di gestione della memoria è capace di supportare più di 1MBdi veloci statiche RMA sincrone and più di 1 memoria FLASH è inclusa. I dati di calibrazione del sensore e del programma sono memorizzati permanentemente nella memoria FLASH e scaricati nella memoria RAM all'avvio del sistema tramite un appropriato circuito di caricamento. Per migliorare la velocità di esecuzione, tutte le operazioni run-time sono poi direttamente eseguite sui dati e sui programmi memorizzati nella RAM, che è anche usata come un buffer. C'è anche un canale per l'accesso diretto in memoria (DMA) per velocizzare lo scaricamento dei blocchi di dati dal vettore di pixel direttamente nella memoria RAM senza l'intervento del processore principale. Un'interfaccia seriale SPI, un prototipo di porta parallela e un'interfaccia video seriale sincrona sono inclusi.

2.3 Il co-processore Totem

Il co-processore Totem a rete neurale ha la struttura di un processore di segnali digitale con 32 processori moltiplicatori/sommatori digitali totalmente paralleli altamente ottimizzati per l'esecuzione in parallelo di una memoria pesato distribuita sul chip (Figura 3). La performance è sull'ordine di 300 M operazioni al secondo di tipo Moltiplicazioni/Addizioni. E' ottimizzato per il calcolo di reti neurali "multy-layer-perceptron" con un numero di strati di neuroni arbitrari, e può eseguire sia in fase di addestramento che di test. La flessibilità del co-processore permette di implementare più di 8000 connessioni. Per esempio, può essere dinamicamente configurato per calcolare una rete neurale con 220 ingressi, uno

strato nascosto con 32 neuroni ed uno strato d'uscita con 42 neuroni in meno di 40 microsecondi. I circuiti furono sviluppati usando una combinazione di metodi di progetto ad alto livello e totalmente dedicati per i blocchi digitali e progetti totalmente dedicati per i vettori dei sensori ottici e blocchi analogici. Alcune tecniche sono state utilizzate per ridurre il rumore introdotto dai circuiti digitali dei fotosensori e blocchi analogici.

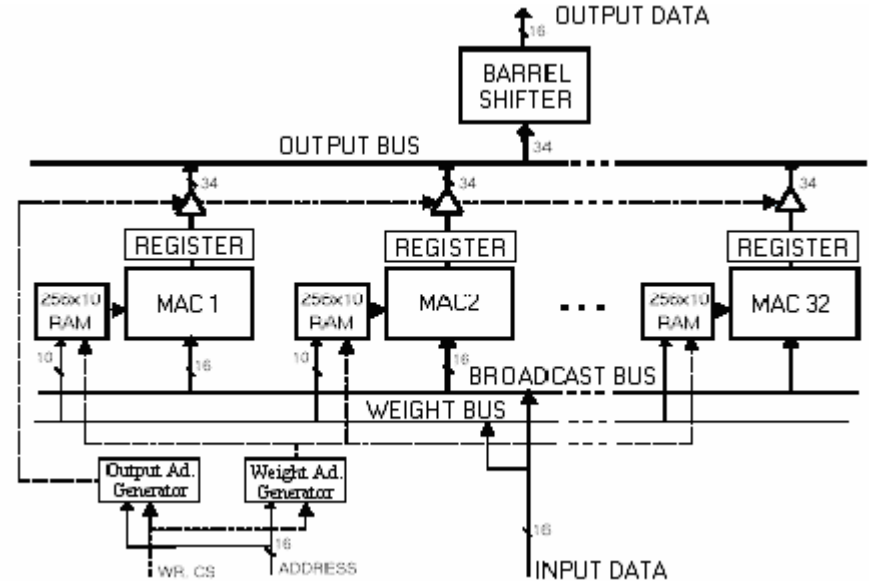

Figura 2.3: Struttura del chip: memorie, connettori e tutti i componenti passivi collegati ad essi.

Capitolo 3

Analisi del Processore TotemRisc per l'Implementazione di Reti Neurali

3.1 Introduzione

Il chip NeuriCam NC3001 è un processore neurale digitale ottimizzato per il calcolo neurale. Questo chip è basato su un'architettura TOTEM la quale integra al suo interno concetti hardware e software per realizzare sofisticate applicazioni di elaborazione dei segnali e provvedere sistemi le cui performance sono calcolate su un miliardo di connessioni (operazioni di moltiplicazioni e somme) per secondo. Questo livello di performance è stato possibile realizzarlo tramite un motore di calcolo parallelo avente un occupazione sul silicio molto ridotta e una bassa dissipazione di potenza.
Dato che il chip è stato ottimizzato per l'esecuzione di prodotti scalati, è molto portato per l'utilizzo con algoritmi di addestramento ed elaborazioni di immagini: in particolare, è provvisto un supporto diretto alle reti neurali concorrenti, a calcoli fuzzy e filtraggi convenzionali. La performance massima di riconoscimento è ottenuta in combinazione con l'algoritmo di addestramento Reactive Tabu Search per le reti neurali.
Il principale obiettivo del chip è creare un nuovo mercato industriale e scientifico abbassando pesantemente il costo per l'OEM ed le applicazioni di controllo e di classificazione per gli utenti finali. A questo scopo il chip è stato incorporato nello sviluppo di schede per essere usato come co-processore nelle workstation e nei personal computer. Il co-processore interagisce col processore principale del sistema host tramite bus standard. Schede con singolo e doppio processore basate su bus PCI, VME e ISA/AT con pieno supporto software sono disponibili e inoltre versioni CompactPCI e PCI/104 sono tutt'ora in sviluppo per uno specifico mercato relativo ai sistemi integrati. La performance della scheda raggiunge 2 miliardi di operazioni al secondo. E' anche stata inserita una interfaccia diretta per la macchina fotografica sulle schede. Inoltre è incluso tutto un ambiente di sviluppo software grafico per poter sviluppare tutto ciò che si vuole.

3.2 Caratteristiche del chip

Il chip NC3001 è un circuito integrato contenente un numero di processori (o "neuroni artificiali") che lavorano in parallelo e sono connessi in modo da minimizzare le operazioni del cervello umano. Esso accetta uno stream di dati (tipicamente dai sensori) sul suo ingresso e lavora su di essi producendo le informazioni di alto-livello contenute nel segnale originale. Questo funzionamento è ottenuto dal corretto "addestrando" del sistema, utilizzando un insieme di esempi noti per aggiustare la configurazione intrna dei processori fino a che essi diano la risposta desiderata.

Il modello di calcolo implementato è robusto rispetto al rumore, alla completezza del segnale e ad altri fattori di disturbo, in maniera tale che la performance di riconoscimento è ampiamente indipendenti da loro. In una tipica applicazione, il segnale elettrico da un traduttore risulta in formato compatto su caratteristiche rilevanti come la similarità di una sequenza a un dato template.

Tutte le operazioni richieste possono essere facilmente eseguite in una sezione semi-automatica o da utenti non esperti con l'aiuto di tool software avanzati. Utenti esperti o sviluppatori di applicazioni possono prendere tutti i vantaggi di un insieme completo di librerie per costruire applicazioni specifiche e dedicate, come nel caso dei sistemi embedded.

Il prodotto ricava il suo livello di performance da due importanti fattori:

1) Un motore di calcolo VLSI pienamente dedicato, regolare e altamente parallelo con semplici unità di elaborazione a virgola fissa e memorie pesate ottimizzate su chip per il calcolo di reti neurali basate su percettroni multi-strato. La sua implementazione digitale assicura un alto livello di realizzabilità, la velocità di calcolo e l'interconnessione altamente ottimizzata provvedono a compattare la dimensione, diminuire la potenza di dissipazione e aumentare la velocità di elaborazione. Le semplici interfacce permettono l'integrazione dell'elaborazione del chip come un co-processore stand-alone o sistemi embedded con un minimo ammontare di componenti esterni.
2) L'innovativo algoritmo di addestramento Reactive tabu Search che è capace di offrire un superiore livello di performance di riconoscimento in combinazione con semplici e piccoli elementi computazionali. Le tecniche di addestramento standard sono anche supportate con lo stesso livello di performance ottenuto con hardware a virgola mobile.

NC3001 – Technical characteristics

- Pipelined Digital Data Stream, Single Instruction Multiple Data (SIMD) architecture
- 32 fixed-point fully-parallel multiply-and-accumulate processors (MACs) operating in parallel from a common broadcast bus. 2's complement data format
- 32 Kbit on-chip dynamic random-access memory organised as 32 blocks of 1Kx8 bits for weight storage with close coupling with processors. Memory can be assigned either to a single neuron or be partitioned among several neurons to implement multi-layer networks with a single chip
- 32-input, 16-output barrel shifter for scaling of results
- Limited word width for economical layout: 16-bit data, 8-bit weight, 16 or 32-bit results
- Performance of 1000 million multiply-and-accumulate operations per second with a 30 MHz clock. A multi-layer perceptron with a 16-16-1 topology can be evaluated in about 2 µs. Higher performance can be achieved by paralleling up to four chips per network level to implement neurons with up to 128 inputs
- Simple interface with data input, data output, memory address and control buses. The chip can operate as a coprocessor in microprocessor systems
- Support circuitry for external look-up table (LUT) RAM to implement activation function of neural network
- Compact chip size (70 mm^2) and extremely limited number of transistors (250.000).

3.3 Algoritmo Reactive Tabu Search

La realizzazione di modelli biologici in sistemi di VLSI altamente concorrenti richiede algoritmi di addestramento che permettono pesi a bassa-precisione, elaborazione dei segnali a bassa accuratezza e solamente la fase di valutazione della rete quando è dato il pattern in ingresso. L'algoritmo di apprendimento Reactive Tabu Search (RTS) soddisfa i sopra elencati requisiti. È caratterizzato sull'ottimizzazione della memoria globale, nessun bisogno di derivate, un corretto numero di bit per i pesi e gli ingressi, robustezza a bassa accuratezza computazionale e nessuna sensibilità critica alle condizioni iniziali. Questo contrasta con algoritmi di addestramento basati sulle derivate come la backpropagation che tende a richiedere calcoli ad alta precisione.

La realizzazione in virgola fissa e con una lunghezza di parole molto piccola, fatta possibile dalla RTS, permette di arrivare ad architetture VLSI più economiche rispetto alle relative in virgola mobile, con formati di parola lunghi in termini di dissipazione di potenza e velocità.
Per il caso particolare di reti neurali concorrenti tempo discreto, i più comuni algoritmi richiedono che la rete sia mappata in una equivalente rete feed-forward usando il meccanismo per esempio della backpropagation. Questo tecnica generalmente da buoni risultati nelle reti con un gran numero di neuroni (spesso dell'ordine di 8-10 volte l'architettura originale RNN) visto che riduce tantissimo la velocità di addestramento e la percentuale di convergenza.

3.4 Ambiti Applicativi

Le macchine basate sul NC3001, molte applicazioni computazionalmente pesanti possono essere eseguite a basso costo, perché la velocità dell'architettura TOTEM supera il livello critico dei calcoli in relazione all'ampiezza, mantendo quindi un basso costo del siste,a. Questo è molto significativo ed importante per l'emulazione dei sistemi, che è spesso troppo bassa per essere usata nelle pratiche reali di molte applicazioni. I sistemi dedicati o integrati costruiscono attorno al processore quindi sistemi a basso costo per soddisfare i livelli del mercato.
Le aree applicative nelle quali il co-processore Totem di NC3001 può essere usato per giungere un livello performance buono sono numerose:

- utenti che operano sulle seguenti applicazioni: controllo di qualità industriale on-line di prodotti flat, terminali per il riconoscimento vocale, sistemi laser per utilizzo medicale, classificazione di eventi fisici, filtraggi e miglioramento di dati in strumenti scientifici per la meteorologia, dispositivi di puntamento ottici, algoritmi di trasmissione e compressione di segnali ECG, analisi di serie temporali, ecc.
- I più significativi mercati disponibili riguardano l'elaborazione dei segnali digitali, analisi e compressione; riconoscimento di voce in ambienti rumorosi per l'industri legata all'automotive; riconoscimento di imponte digitali e di volti per i sistemi di validazione; sistemi di controllo del tracciato delle strade, sistemi di controllo per l'accesso nelle stazioni di trasporti pubblici; compressioni di segnali biomedici; elaborazione di segnali per strumenti scientifici. Alcuni di questi potenziali mercati

sono attivamente in collaborazione con compagnie e laboratori di ricerca per prendere tutti i possibili vantaggi di questo approccio nell'area di riconosciemtni di pattern e processori paralleli.

- Altri potenziali mercati sono i sistemi di supporto per i disabili, riconoscimento di caratteri ottici; elaborazione di segnali per videoconferenze, telemedicina, controllo di veicoli, e controllo di ambienti aperti.

3.5 Vantaggi di Utilizzo

I vantaggi dell'architettura sono elencati sotto:

- velocità di riconoscimento estremamente alta;
- addestramento robusto e veloce di reti neurali;
- vantaggio in termine di rapporto performance/prezzo;
- rapporto estremamente buono potenza/potenza dissipata, ideale per sistemi embedded;
- impegno costante per una continua evoluzione del prodotto tramite team di sviluppo per ciò che riguarda reti neurali, VLSI, sistemi hardware e software.

Un recente studio eseguito dal laboratorio KTH Istituto Reale della Tecnologia, Stoccolma la Svezia mostra che l'architettura TOTEM riesce ad eseguire prodotti simili all'INTEL, IBM ed altri in certi e particolari campi di studio.

3.6 Cenni sull'evoluzione del prodotto

Lo sviluppo dell'ambiente teoretica del microprocessore e degli algoritmi iniziò nel 1990 come una collaborazione tra due ricercatori dell'IRST e dell'Università di Trento. Lo sviluppo dell'architettura hardware continuò nel 1993 con un gruppo di tre ricercatori. Il progetto sul silicio iniziò nel 1994 con contributi dall'IRST, delle Università di Trento e Kent a Canterbury e dell'Istituto Nazionale italiano per la Fisica Nucleare INFN. Il primo prodotto al silicio completamente funzionale fu prodotto nel settembre 1994 all'interno del progetto dell'EU Esprit Project 7101 MInOSS ed INFN. Nei successivi tre anni furono sviluppati

versioni migliorate del chip, a processore singolo PC ISA, e le relative schede VME furono esaminati e rilasciate ad utenti selezionati con la prima versione dei driver software ed un numero di applicazioni furono sviluppate nei vari istituti di ricerca e università. Nel maggio 1998, NeuriCam S.r.l. fu implementato a Trento con lo scopo di portarlo sul mercato e sviluppare le sue ulteriori versioni aggiornate.

3.7 Possibili Sviluppi futuri

Il progetto NeuriCam ha pianificato di intraprendere lo sviluppo di nuove versioni dell'architettura Totem con nuovi, più piccoli e più veloci processori che offrono maggior compatibilità con versioni più nuove con i sistemi software presenti. L'innovazione contenuta nei nuovi prodotti è basata sulle migliori tecnologie di fabbricazione ed architetture. In particolare sono in fase di sviluppo i seguenti prodotti:

- una versione del microprocessore con maggiori (256 byte per neurone) memorie pesate ed una frequenza di lavoro di 50 MHz;
- una versione del processore di 128-neuroni con 256-byte per i pesi della memoria ed un livello di performance di 5 GCPS;
- una rete neurale concorrente tempo discreto sul chip per permettere l'efficiente implementazione dei sistemi di riconoscimento per segnali variabili nel tempo come quelli incontrati nei sistemi di riconoscimento del parlato;
- moduli software capaci sia di nascondere i dettagli hardware all'utente che di provvedere alti livelli di astrazione nella descrizione dei blocchi computazionali;
- schede e circuiti integrati contenenti una CPU per applicazioni embedded stand-alone;
- un computer neurale basato su CPD standard e bus CompactPCI.

Una nuova architettura, chiamata TOTEM++, è stata introdotta per l'uso di calcoli approssimati. Questo costituisce un avanzamento notevole nel calcolo neurale perché si avvicina alla semplicità della realizzazione analogica mantenendo i vantaggi della realizzazione digitale: affidabilità, scalabilità con nuove tecnologie, facilmente assimilabili dagli utenti. Le prove realizzate su casi reali mostrano che una performance eccellente può essere ottenuta da reti neurali operanti sul principio di calcoli approssimati dato che la rete è capace di adattarsi alla sua propria risposta approssimata. In pratica, questo è

realizzato tramite un adattamento automatico dei pesi durante la fase di apprendimento. La nuova architettura si prevede che produca almeno il doppio in performance con un significativo abbassamento di potenza e aera occupata di silicio.

BIBLIOGRAFIA

[1] R. Battiti, P. Lee, A. Sartori, G. Tecchiolli, “TOTEM: a Digital Processor for Neural Networks and Reactive Tabu Search”, MICRONEURO 94.

[2] NeuriCam’s NC3003 Datasheet: TOTEM Digital Processor for Neural Networks, Rel. 12/99

[3] NeuriCam’s Number Plate Recognition System (www.neuricam.com)

[4] R. Battiti, A. Sartori, G. Tecchiolli, P. Tonella, A. Zorat, “Neural Compression: an Integrated Application to EEG Signals”, IWANT95.

[5] Zorat, A. Satori, G. Tecchiolli, L. Koczky, “A Flexible VLSI Processor for Fast Neural Network and Fuzzy Control Implementation”, LIZUKA96.

[6] NeuriCam’s Application Note AN005, “Using the NC3001 for DSP Applications: computing the DFT of a 256x256 image”.

[7] NeuriCam’s Parallel Signal Processing Boards: TOTEM PCI Technical Reference Manual (Rel. 12/99)

[8] Xilinx: The Programmable Logic Data Book 2001.

[9] Peter Alfke, “Evolution, Revolution and Convolution. Recent Progress in Field- Programmable Logic”, FPL2001.

[10] Xilinx: Spartan ASIC Alternatives.

[11] S. Dusini et al, “The Neurochip Totem in the Higgs Search”, ABANO96.

[12] L. Ricci, G Tecchiolli, “A Neural System for Frequency Control of Tunable Laser Sources”, IEEE IMTC'97.

[13] F. De Nittis, G Tecchiolli, A. Zorat, “Consumer Loan Classification Using Artificial Neural Networks”, EIS'98.

[14] R. Battiti and G. Tecchiolli, «Training neural nets with the reactive tabu search», IEEE Transactions on Neural Networks, 1995.

[15] T. Nordström and B. Svensson, «Using and designing massively parallel computers for artificial neural networks», Journal of Parallel and Distributed Computing, vol. 14, No. 3, 1992, pp. 260-285.

[16] A. Mazzetti, Reti neurali artificiali, Apogeo Editore, 1992

[17] B. Mueller, J. Reinhardt: Neural Networks Springer-Verlag, Berlin,1990

[18] R. Lippmann, An Introduction to Computing with Neural Nets, IEEE ASSP MAGAZINE, Aprile, 1987.

Appendice A

A.1 Descrizione Dettagliata del Chip

Il progetto hardware di reti neurali altamente parallele richiede cura nel progetto dell'architettura e dell'ottimizzazione. L'architettura TOTEM utilizza una struttura digitale SIMD che permette di moltiplicare gli elementi di elaborazione di base (PE o neuroni) per essere usati con un elevato grado di parallelismo. La semplicità architetturale complessiva è realizzata ottimizzando l'esecuzione delle funzioni di moltiplicazione e addizione su stream di dati long, le operazioni più complesse e computazionalmente pesanti per questo tipo di reti.

L'architettura generica del processore usata per implementare il percettrone multi-strato di contiene un vettore di N_{neu} di elementi di elaborazione parallela (i neuroni) connessi insieme tramite un broadcat bus che è usato per introdurre le caratteristiche nei neuroni e leggere i risultati di uscita dai neuroni stessi. Questo porta la desiderata totale connettività nella rete MLP utilizzando il livello più basso possibile della banda nella porta d'ingresso di un trasferimento I/O per un ciclo di clock.

Ogni neurone contiene un'unità di moltiplicazione e accumulatore ed un registro per il risultato, per permettere direttamente l'implementazione di reti multi-strato. L'elaborazione efficiente è realizzata da un'architettura di moltiplicatori totalmente parallela, che offre elevata velocità di un'operazione per ciclo di clock, mentre il tempo richiesto per un completo moltiplicazione-addizione è dell'ordine di N_{inp} cicli di clock, dove N_{inp} è il numero di caratteristiche d'ingresso.

La larghezza di banda richiesta per fornire i pesi al MAC durante la fase di addestrando è dell'ordine di N_{neu} operazioni di input/output operazioni per ciclo, che può causare un collo di bottiglia tra l'I/O con la memoria usata. Questo problema è stato risolto in Totem utilizzando la memoria interna per i pesi. Questo permette una banda larga tra MAC e memoria facendo aumentare tuttavia l'area di silicio. Il poco frequente calcolo dei cambi dei pesi durante la fase di addestramento è delegato ad altri circuiti nel sistema, come il processore dell' host. Speciali blocchi per generare indirizzi per i pesi in memoria e i registri provvedono ad offrire una maggior flessibilità per il calcolo di regole fuzzy, e filtri FIR e IIR. Questo approccio è stato provvisto per offrire un eccellente performance

quando sono applicati insieme all'algoritmo di addestramento Reactive Tabu Search, ma è andato bene anche con altri algoritmi.

Un comune shifter 32-16 bit è stato inserito nel canale d'uscita per scalare i risultati e convertirli a lunghezza word per eventuali e futuri utilizzi negli strati della rete neurale.

La funzione di attivazione è implementata tramite una veloce memoria LUT. Questo approccio è stato scelto perché offre la massima flessibilità riducendo inoltre l'area di silicio complessiva del circuito. La paginazione della memoria può essere usata inoltre per immagazzinare funzioni di attivazione diverse. Una minima configurazione dell'architettura che usa solo uno frammento è mostrata in Figura. 3.

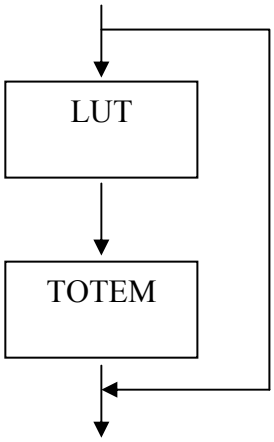

Figura A.1: La rete neurale implementata con il processore NC3001: configurazione minima.

L'approccio modulare usato nel progetto abilita potenti architetture combinando un numero di chip operanti in parallelo, Figura. 4.

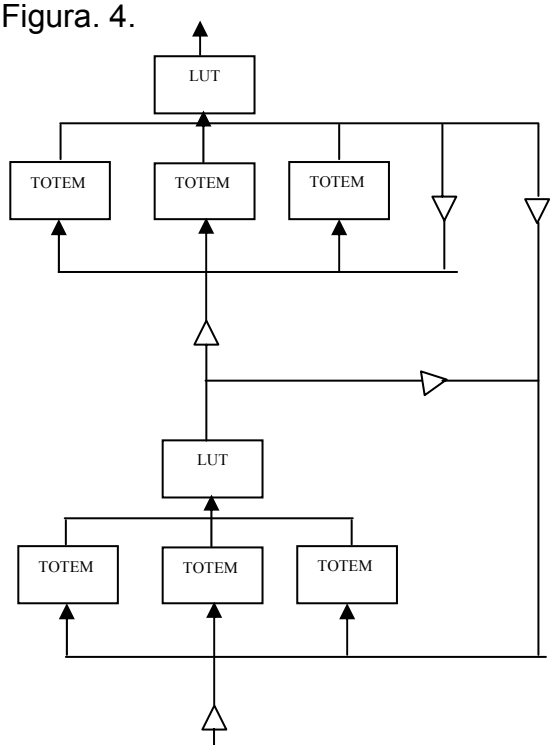

Figura A.2: La rete neurale implementata con il processore NC3001: configurazione multiprocessore.

L'incremento della performance del sistema è una funzione lineare del numero di processori usati.

L'architettura parallela descritta sopra è integrata nel processore NC3001 come un vettore di 32 unità di elaborazione parallela con associate memorie e logica di controllo. Le ampiezze delle parole sono ottimizzare per imparare con l'algoritmo RTS: l'ampiezza di 16 bit con il bus broadcast è adeguata per rappresentare segnali e trasduttori e risultati intermedi tra gli strati. La memoria con ampiezza i parola pari a 8 bit è sufficiente per molti lavori di classificazione e permette chip anche commerciali. L'ampiezza di parola a 32 bit del canale d'uscita permette un'alta capacità di accumulazione. I vincoli del progetto di memorie comptatte e celle di moltiplicatori è stato soddisfatto tramite progetto perfettamente integrato di questi elementi e tramite anche un confronto di fattori di forma della memoria e blocchi MAC.

L'elemento di base PE è usato per implementare i neuroni contenenti due elementi principali: il moltiplicatore-sommatore (MAC) e a memoria. Il MAC opera sul complemento a due di interi segnati. Il moltiplicatore parallelo 16X8, figura 5, usa l'algoritmo di Baugh-Wooley che, per questi dimensioni di moltiplicatore, provvede ad una configurazione di strato più regolare rispetto a quella ampiamente usata dal moltiplicatore di Booth.

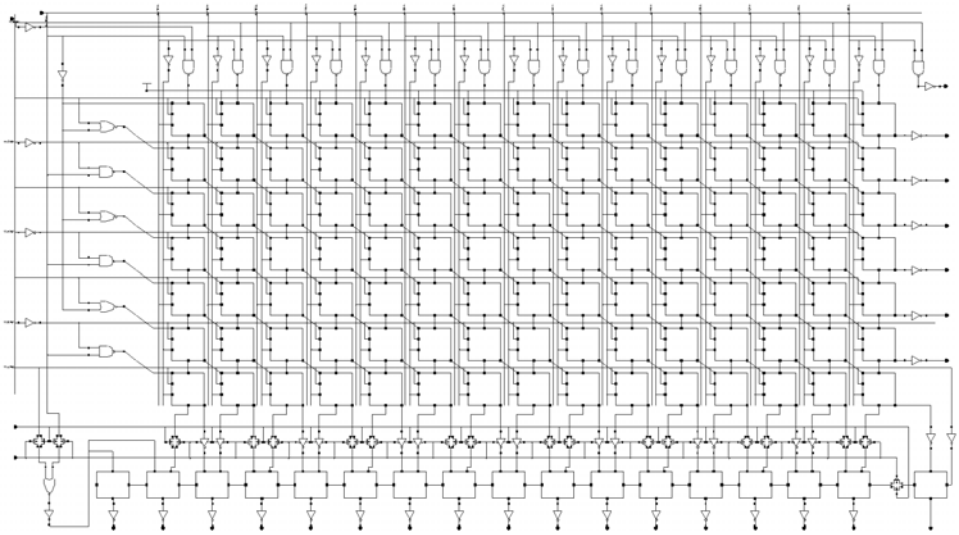

Figura A.3: Moltiplicatore con pipeline 16 x 8 2 di Baugh-Wooley

La cella di base del moltiplicatore contiene 28 transistor statici CMOS. Per operazioni ad alta velocità, è stato incluso un livello di pipeline nel moltiplicatore vettoriale. Le dimensioni del moltiplicatore vettoriale sono 0.84X0.24 mm. L'accumulatore a 32 bit usa lo stesso sommatore usato nel moltiplicatore vettoriale ma con due livelli addizionali di pipeline per

assicurare operazioni ad alta velocità. La pipeline fu usata rispetto configurazioni del sommatore tipo carry-lookahead o carry-saved per la regolarità e la compattezza del circuito.
Una memoria ad accesso casuale dinamica è stata usata per i pesi al fine di provvedere ad una maggior capacità di rimaneggiamento durante la fase di addestramento della rete. Nella fase operativa in modo riconoscimento la memoria non volatile dovrebbe essere sufficiente.
La dimensione di memoria richiesta per uno strato della rete è uguale al massimo numero di neuroni di ingresso che dovrebbe essere implementato. Nella presente architettura, un banco di memoria dedicato on-chip è stato associato ad ogni neurone. Grazie alla flessibilità del circuito di generazione di indirizzi, il chip può implementare differenti topologie di MLP e RNN con un banco di memoria assegnato per ogni neurone oppure suddiviso tra più neuroni su differenti strati. L'ampiezza della memoria è direttamente legata alla precisione richiesta durante l'addestramento: una parola con ampiezza 8 bit è stata scelta come un buon compromesso tra memorie di piccole dimensioni e idoneità agli algoritmi di addestramento e molte applicazioni di filtraggio di segnali (parola di 4-bit di ampiezza dovrebbero essere sufficienti per gli scopi degli RTS). La memoria è stata suddivisa in blocchi di 2-Kbit e accoppiate ai loro relativi processori. Dato che l'area delle memorie tende a prevalere nel chip in questa architettura, è stat usata la memoria RAM dinamica. Area limitata relativa alla RAM è stata realizzata tramite l'uso di celle dinamiche n-MOS. La capacità di tenuta delle celle di memoria è 15 fF. Questa struttura di memoria offre un buon compromesso tra occupazione di area, consumo corrente e semplicità di progetto. Le operazioni di lettura sono eseguite in un singolo ciclo di clock, mentre una richiesta di scrittura in due cicli. La capacità di lettura multipla permette di essere implementata tramite un semplice schema di refresh. Il refresh di una colonna completa di celle richiede due cicli di clock che traslano in un overhead di refresh del 2% circa (assumendo 30s il periodo di refresh). L'uscita della RAM è in pipeline per nascondere il tempo di accesso della memoria e quindi ottimizzare il grado di performance della velocità. Per minimizzare l'area delle righe dei decoder per le memorie furono condivise degli stack di vettori di blocchi di memoria. La dimensione di di 8 bit delle parole della memoria, di conseguenza una dimensione di memoria pari a 256, permette di implementare più di 256 neuroni di ingresso senza accedere a memorie esterne.
Per operazione molto veloci e per assicurare un semplice interfacciamento con i dispositivi esterni, sono stati aggiunti un piccolo numero di fasi in pipeline all'unità MAC e al bus

d’uscita. Il tempo di latenza totale attraverso l'unità di MAC è di due clock. Un registro supplementare per la pipeline è stato aggiunto nella fase d’uscita per sincronizzare l’uscita dell’unità MAC dopo l’operazione di shift ma prima del pad d’uscita. Nella seguente tabella si possono vedere tutte le caratteristiche del chip considerato.

Table 1. Characteristics of the processor chip

Operating frequency	30 MHz
Performance	1 GCPS
Memory size	32 Kbits
Memory access time	< 10 ns
Memory retention time	80 ms @ 80 °C junction 70 mm_2
Die size	
Transistor count	250.000
Power dissipation	2 W @ 30 MHz, 5 V

Tabella A.1: Caratteristiche del chip