

## Realizzazione di un layer convoluzionario di una Convolutional Neural Network

Una Convolutional Neural Network (CNN) è un tipo di rete neurale tipicamente utilizzata per applicazioni di riconoscimento vocale, (ad esempio Keyword spotting system) riconoscimento di oggetti etc.

Un layer convoluzionario è la parte di una CNN in cui si effettua l'operazione di convoluzione tra un filtro  $f$  e una matrice d'ingresso.

In questo caso, il layer convoluzionario è costituito da:

- 1 canale d'ingresso  $C_{in}$ .
- N canali d'uscita  $C_{out}$ . Ad ogni canale di uscita è associato un filtro  $f$  di dimensioni  $f_h \times f_w$ .

Ogni elemento di un  $C_{out}$  è il risultato della convoluzione tra  $f_h \times f_w$  elementi del filtro ad esso associate e  $f_h \times f_w$  elementi della matrice d'ingresso come mostrato nell'equazione 1:

$$conv = \sum f_i \cdot c_i \quad (1)$$

Un esempio di convoluzione è mostrato in figura 1. Partendo dalla situazione iniziale indicata in figura 1.a, per calcolare gli elementi successivi si sposta il filtro verso destra di un elemento alla volta fino a raggiungere la colonna finale della matrice d'ingresso come mostrato in figura 1.b. Per calcolare gli elementi successivi del  $C_{out}(i)$  è necessario spostare il filtro nella riga successiva come evidenziato in figura 1.c. Procedere in questo modo fino a scorrere tutti gli elementi della matrice di ingresso e raggiungere la posizione evidenziata in figura 1.d.

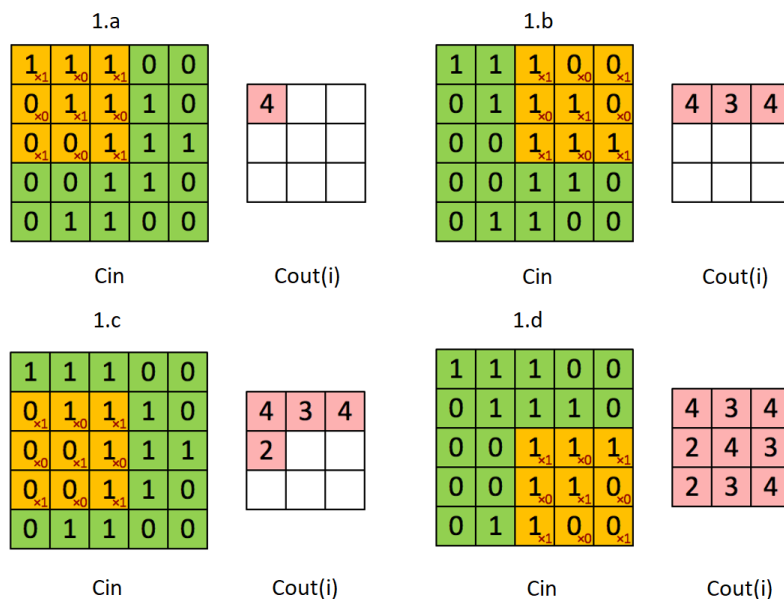


Figura 1

Nel caso specifico, realizzare un layer convoluzionario in cui viene calcolato un elemento di un canale di uno dei canali di uscita  $C_{out}(i)$  per ciclo di clock. Salvare i dati parziali in una memoria.

Utilizzare parametri generici là dove è possibile

La relazione finale del progetto deve contenere:

- Introduzione (descrizione algoritmo, possibili applicazioni, possibili architetture, etc.)
- Descrizione dell'architettura selezionata per la realizzazione (diagramma a blocchi, ingressi/uscite, etc.)
- Codice VHDL (con commenti dettagliati)
- Test-plan e relativi Testbench per la verifica
- Risultati della sintesi logica automatica su piattaforma Xilinx FPGA Zynq: risorse utilizzate (slice, LUT, etc.), massima frequenza di funzionamento, cammino critico, etc. commentando eventuali messaggi di warnings.
- Conclusioni