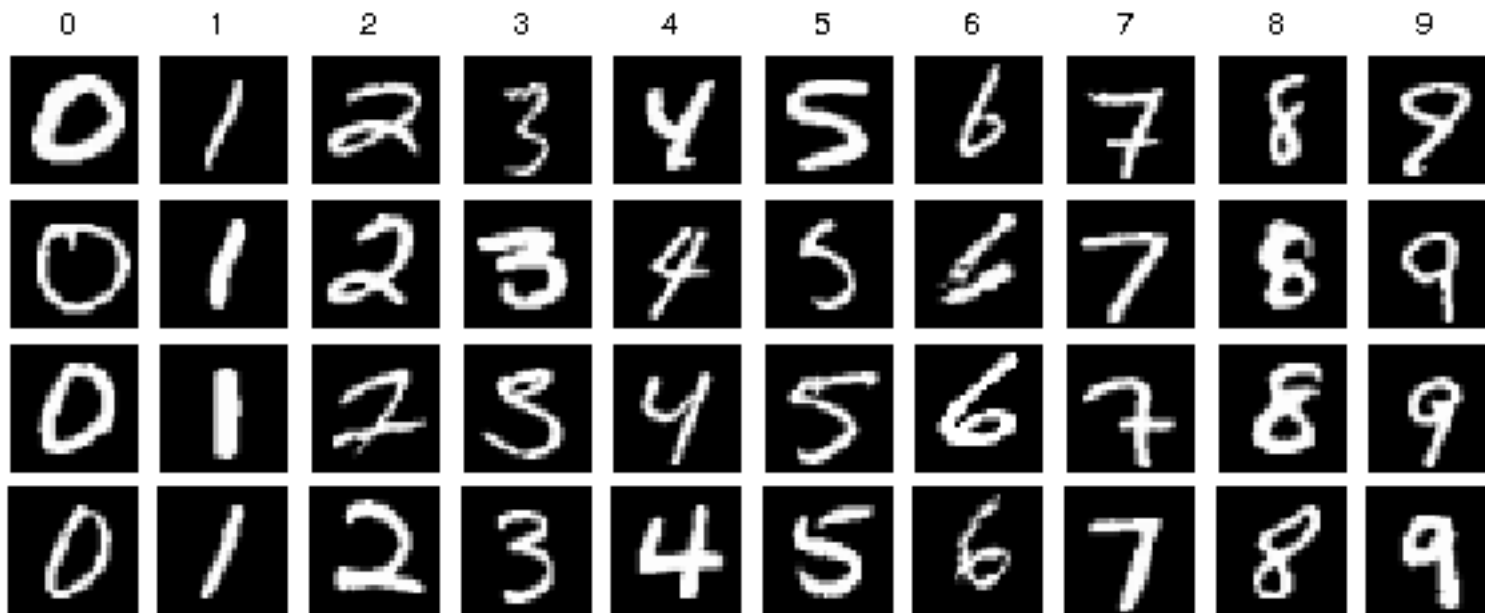


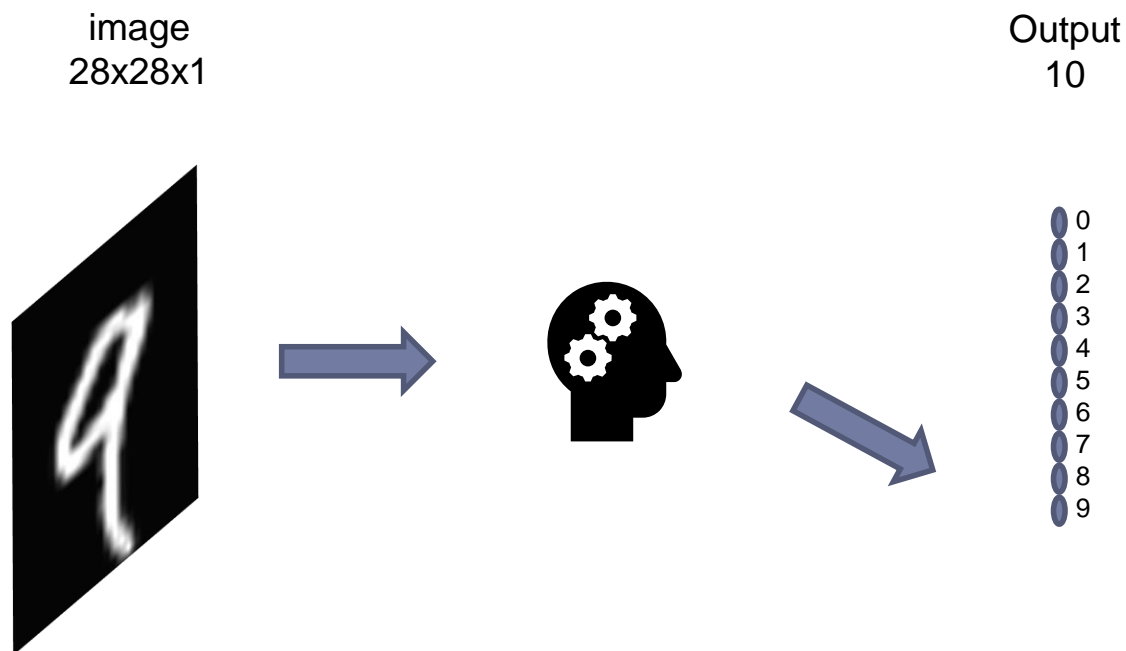
CLASSIFICAZIONE CON LENET

MNIST handwritten digits recognition

- Il MNIST database contiene 60.000 immagini di training e 10.000 di test di 28 x 28 pixel



Riconoscimento di MNIST con LeNet

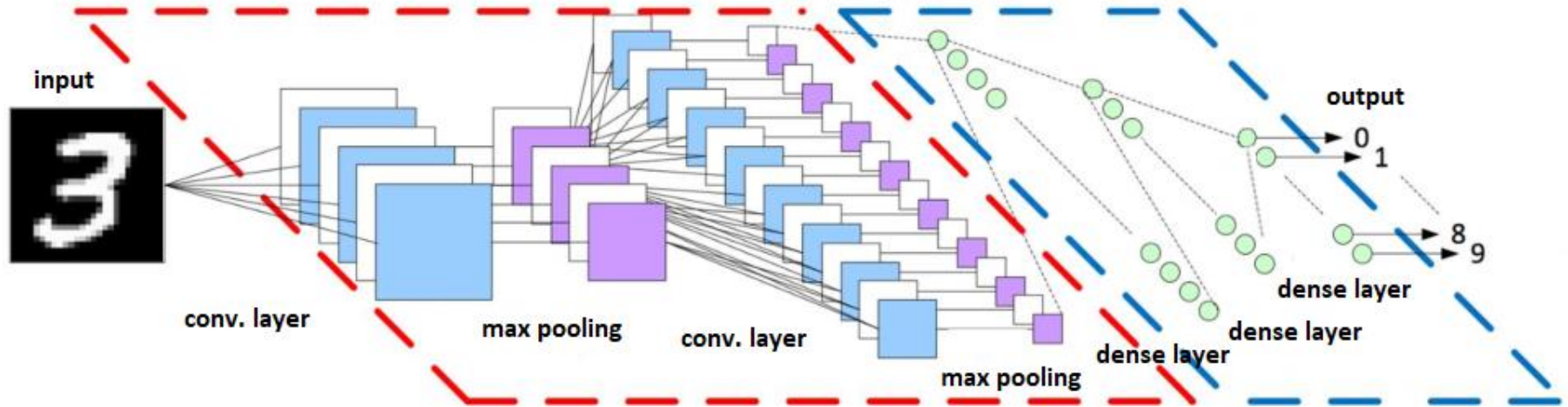


Passi

1. Definizione dell'architettura LeNet;
2. Caricamento e preparazione dei dati MNIST;
3. Addestramento;
4. Analisi delle prestazioni.

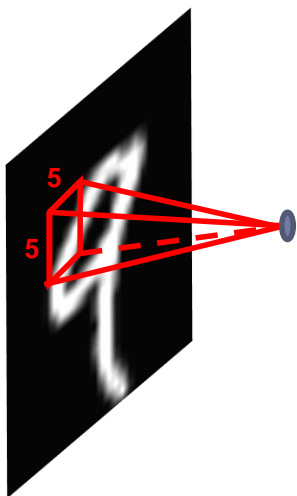
Architettura LeNet

- Consiste di due parti:
 - un blocco di strati convoluzionali (per l'estrazione delle feature)
 - un blocco di strati fully connected (per la classificazione)



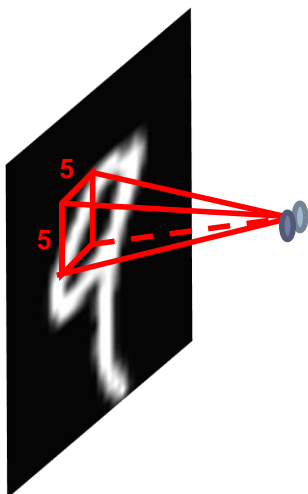
Architettura LeNet

image
28x28x1



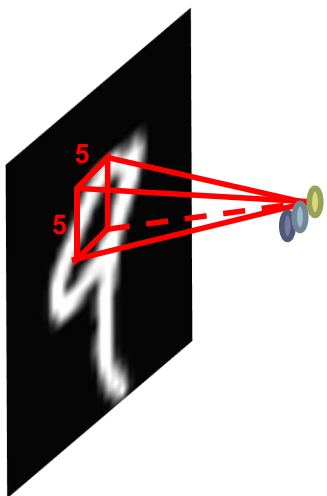
Architettura LeNet

image
28x28x1

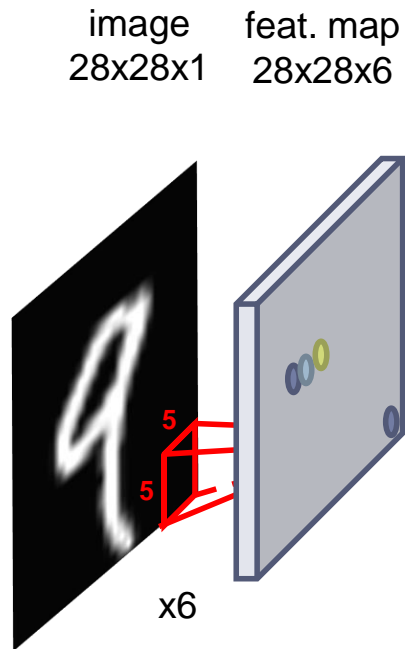


Architettura LeNet

image
28x28x1

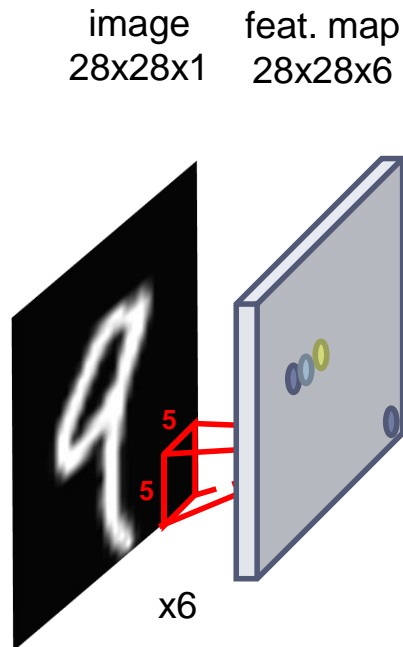


Architettura LeNet



$$F(i, j, c) = b_c + \sum_m \sum_n \sum_k I(i + m, j + n, k) w_c(m, n, k) \quad c = 0, \dots, 5$$

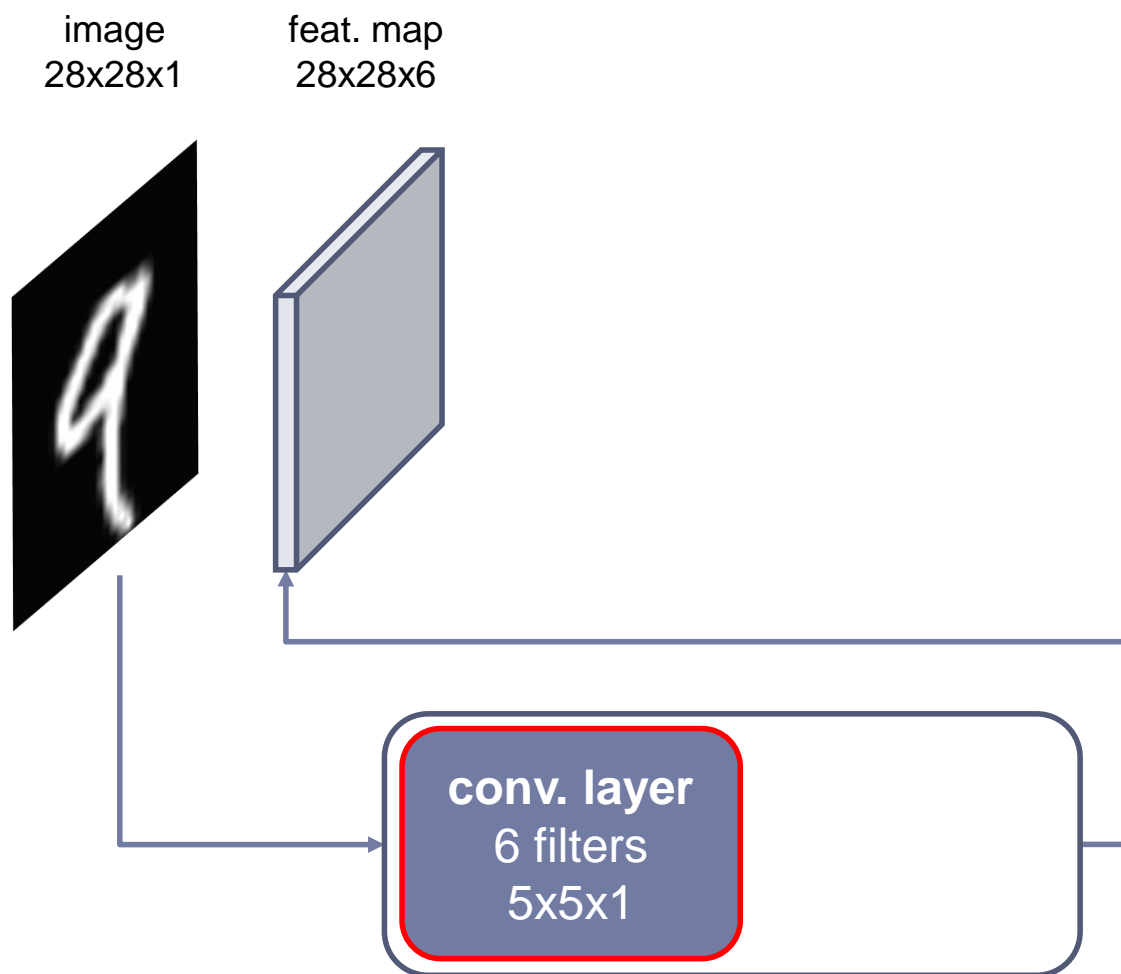
Architettura LeNet



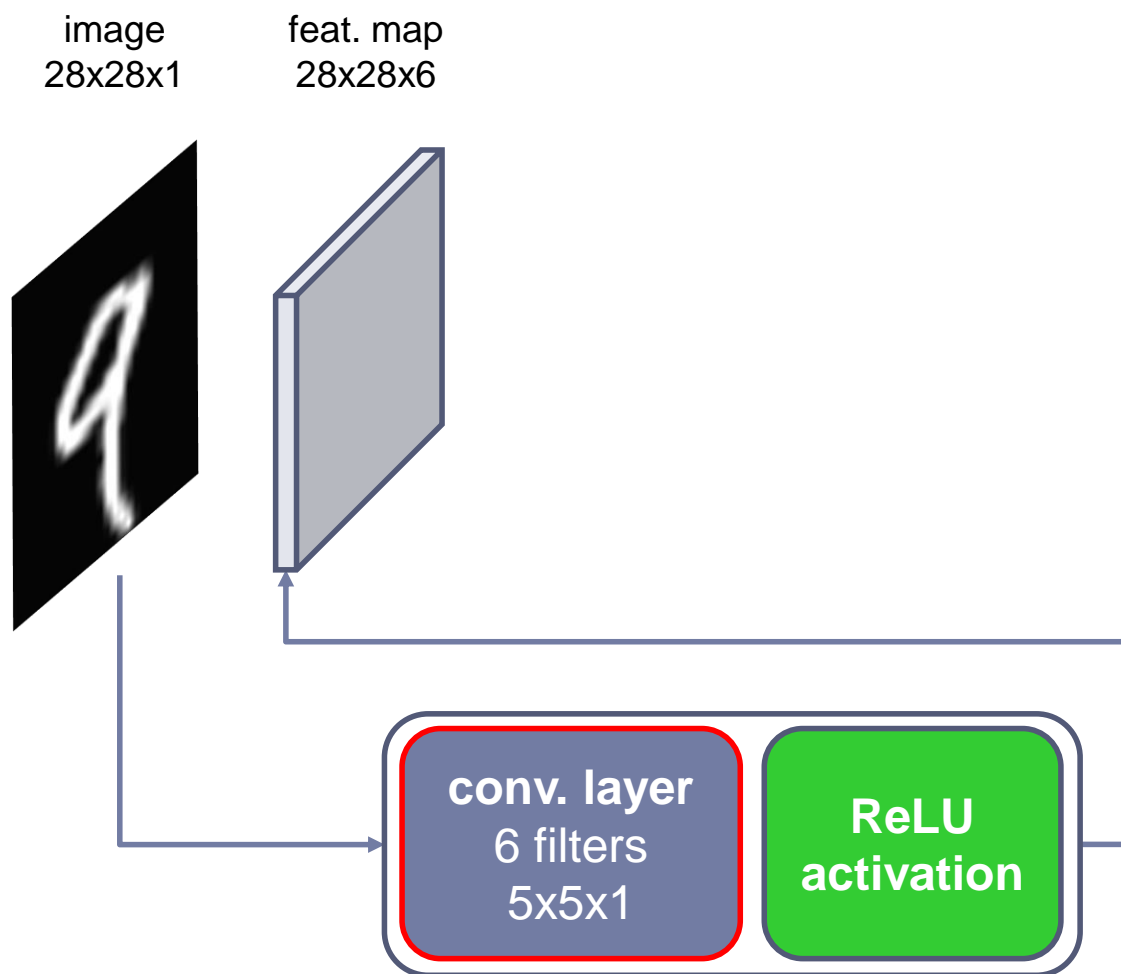
$$F(i, j, c) = \underbrace{b_c}_{6} + \sum_m \sum_n \sum_k I(i + m, j + n, k) \underbrace{w_c(m, n, k)}_{5 \times 5 \times 1 \times 6} \quad c = 0, \dots, 5$$

156 parameters

Architettura LeNet



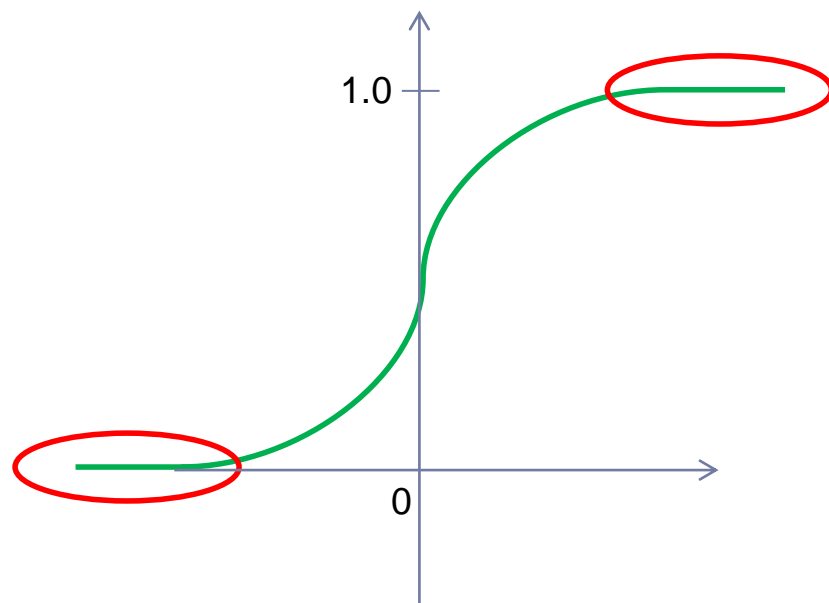
Architettura LeNet



Funzione di attivazione: sigmoide

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

- Una delle prime funzioni adottate
- La zona di saturazione (in rosso) causa i **vanishing gradients**
- L'output è in $[0, 1]$, quindi non è centrato in zero, il che assicura un training più stabile



Vanishing gradients: durante la backpropagation i gradient si avvicinano a zero \rightarrow i pesi non si aggiornano!

Questo è un problema per reti molto profonde

Funzione di attivazione: ReLU

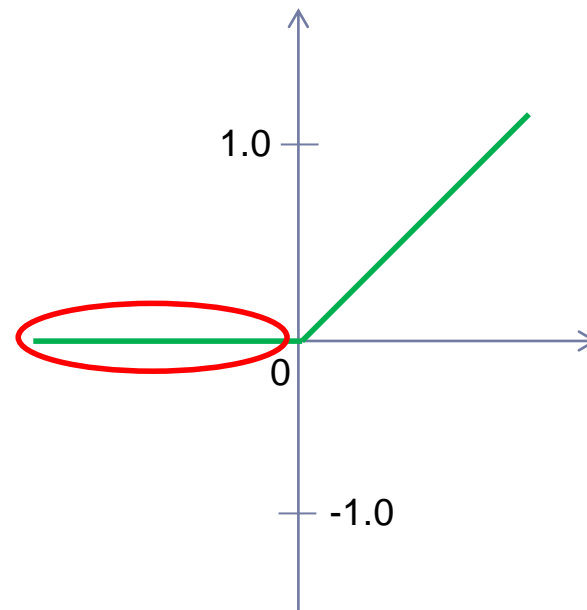
$$\varphi(x) = \max(0, x)$$

Pros.

- Rapida convergenza
(6x w.r.t. sigmoid/tanh)

Cons.

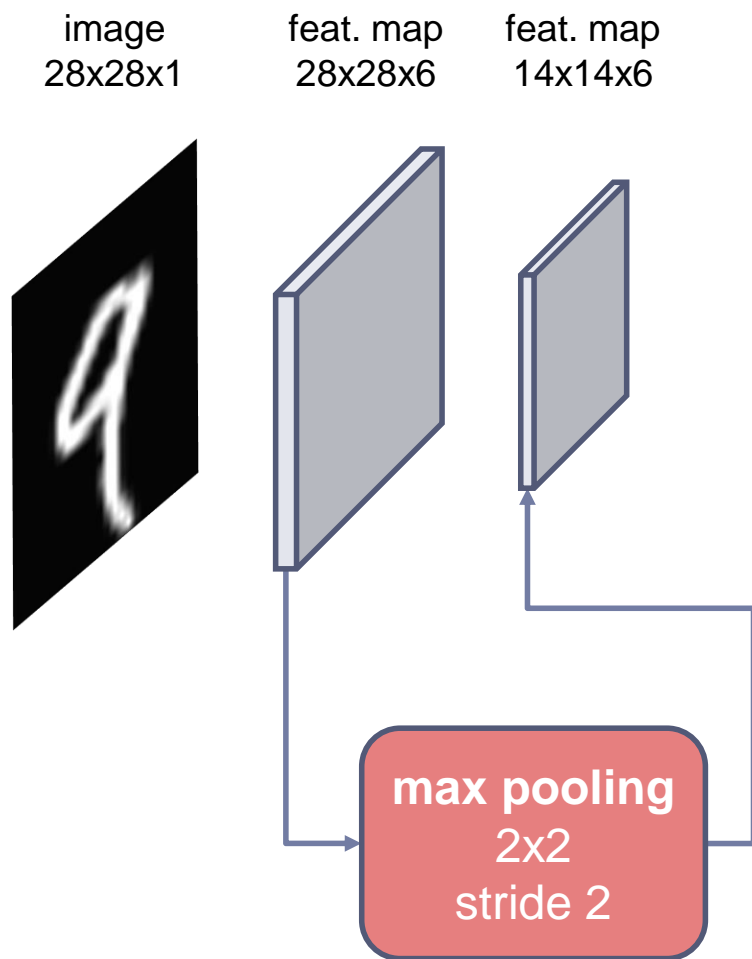
- I neuroni possono non dare alcun contributo



Durante la fase di forward, l'uscita del neurone è uguale a zero e quindi la sua derivata è pari a zero durante la fase di backward

→ Il neurone non fornisce alcun contributo e i suoi pesi non cambiano

Architettura LeNet



Max pooling

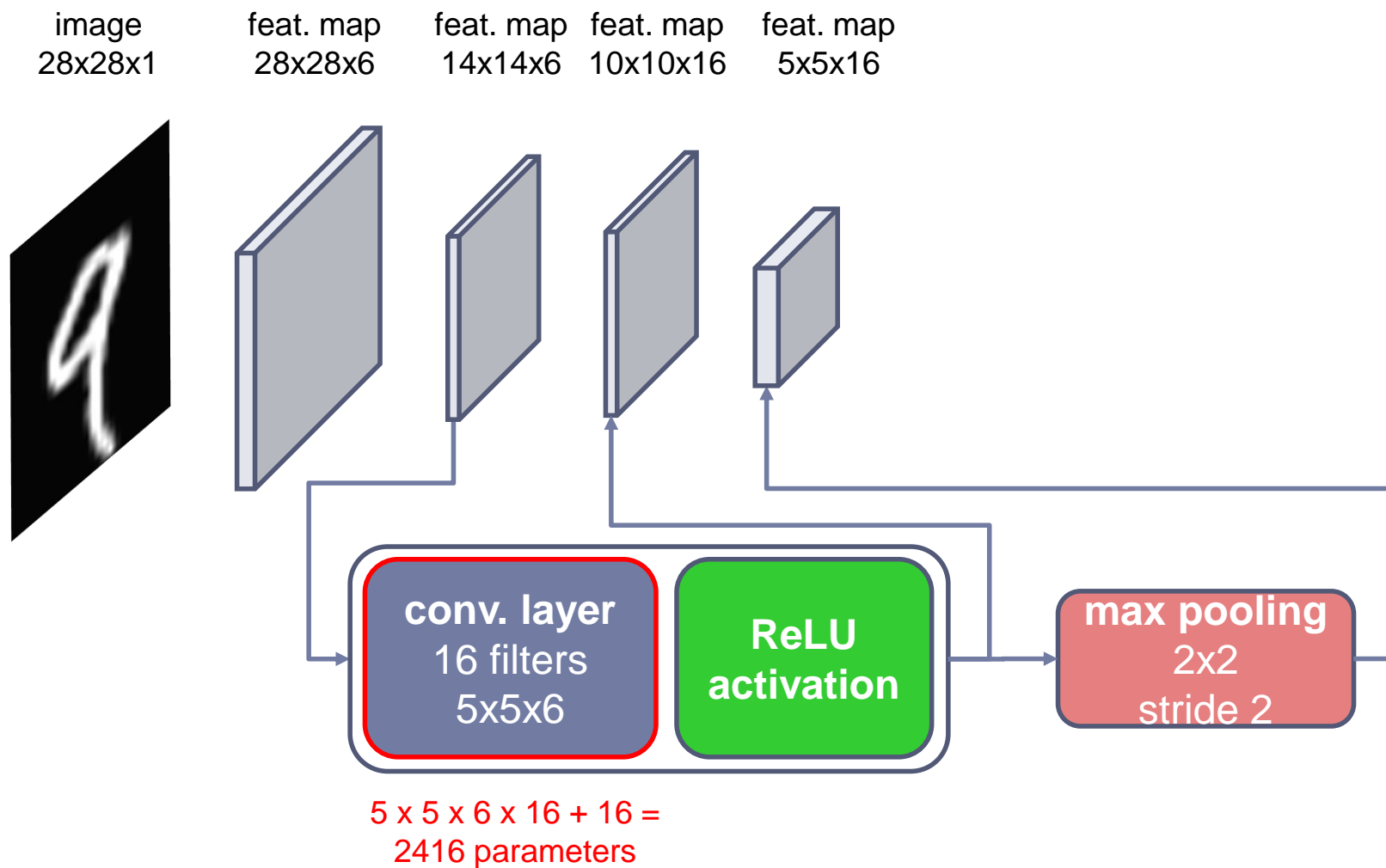
before pooling

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

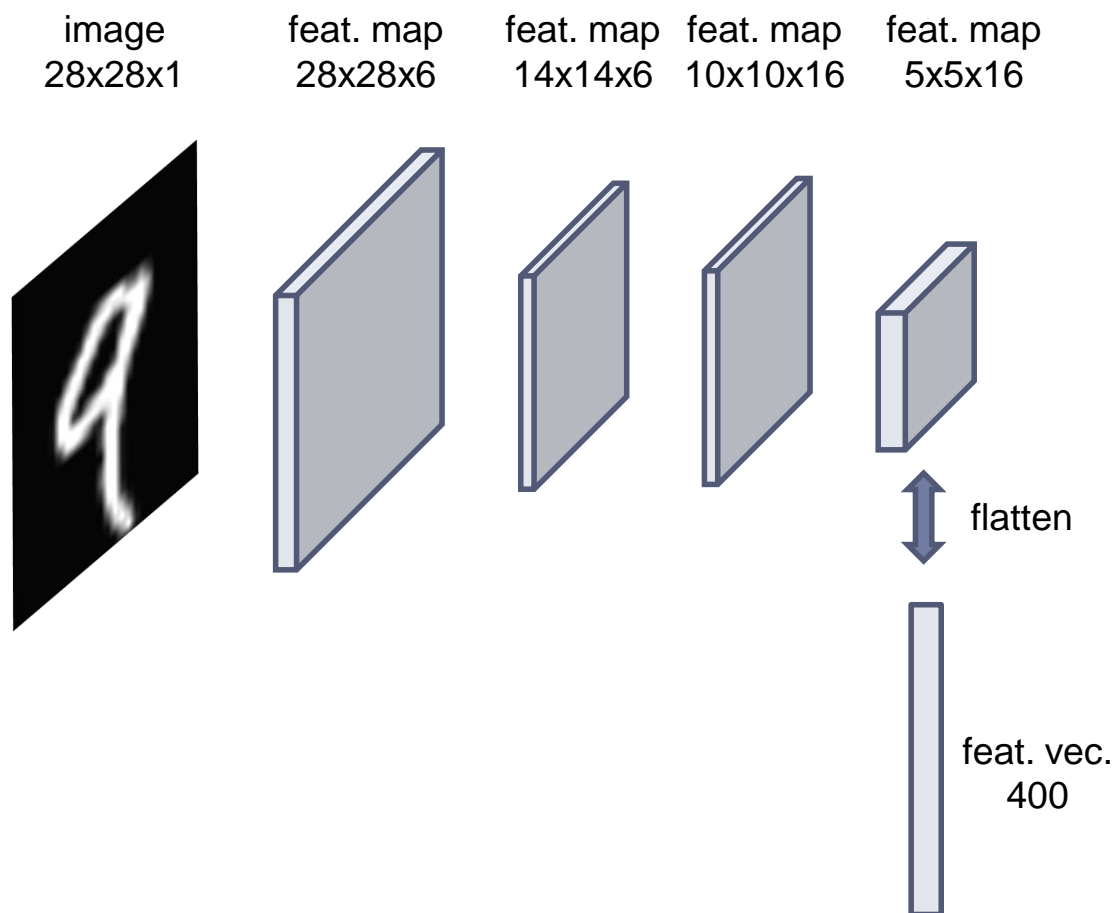
after max pooling
2x2, stride 2

20	30
112	37

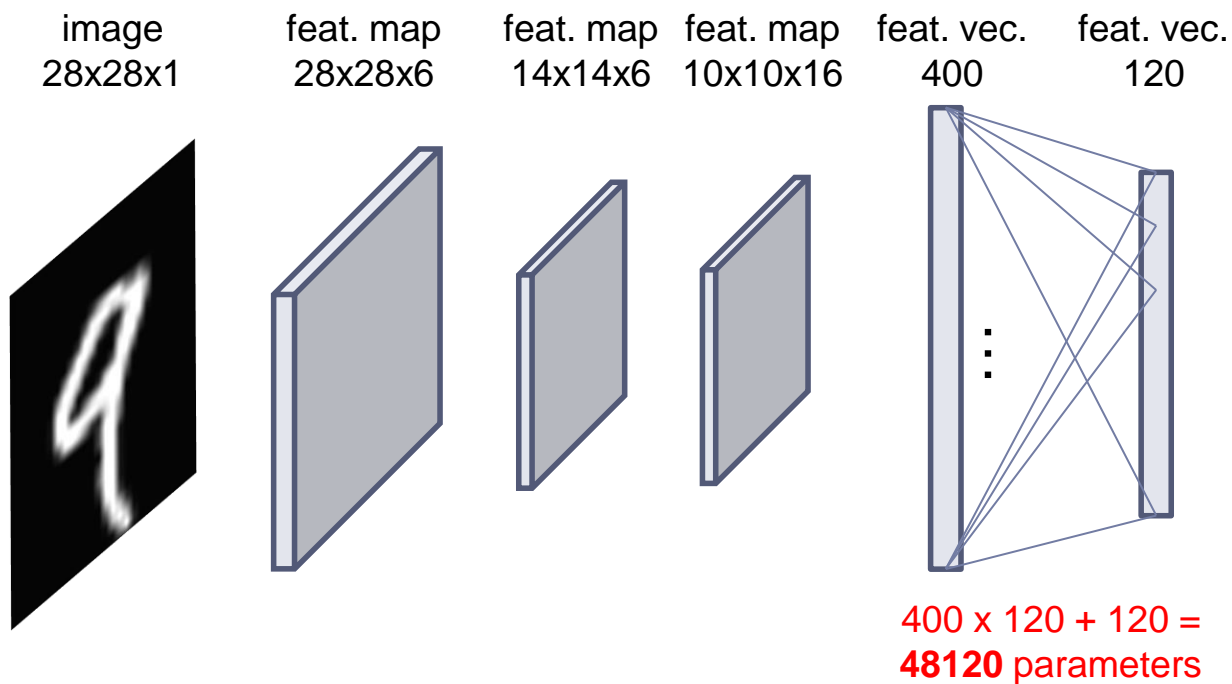
Architettura LeNet



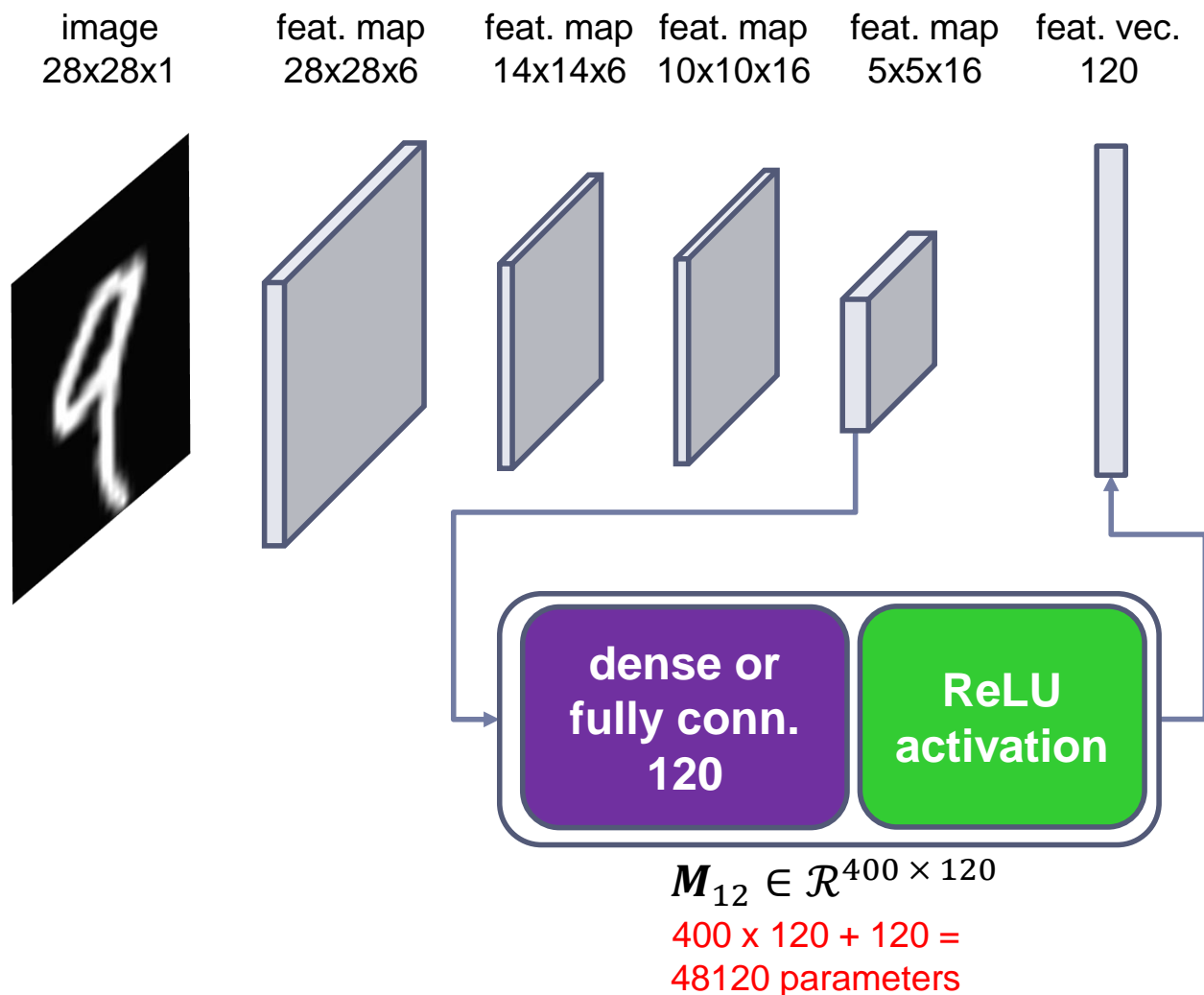
Architettura LeNet



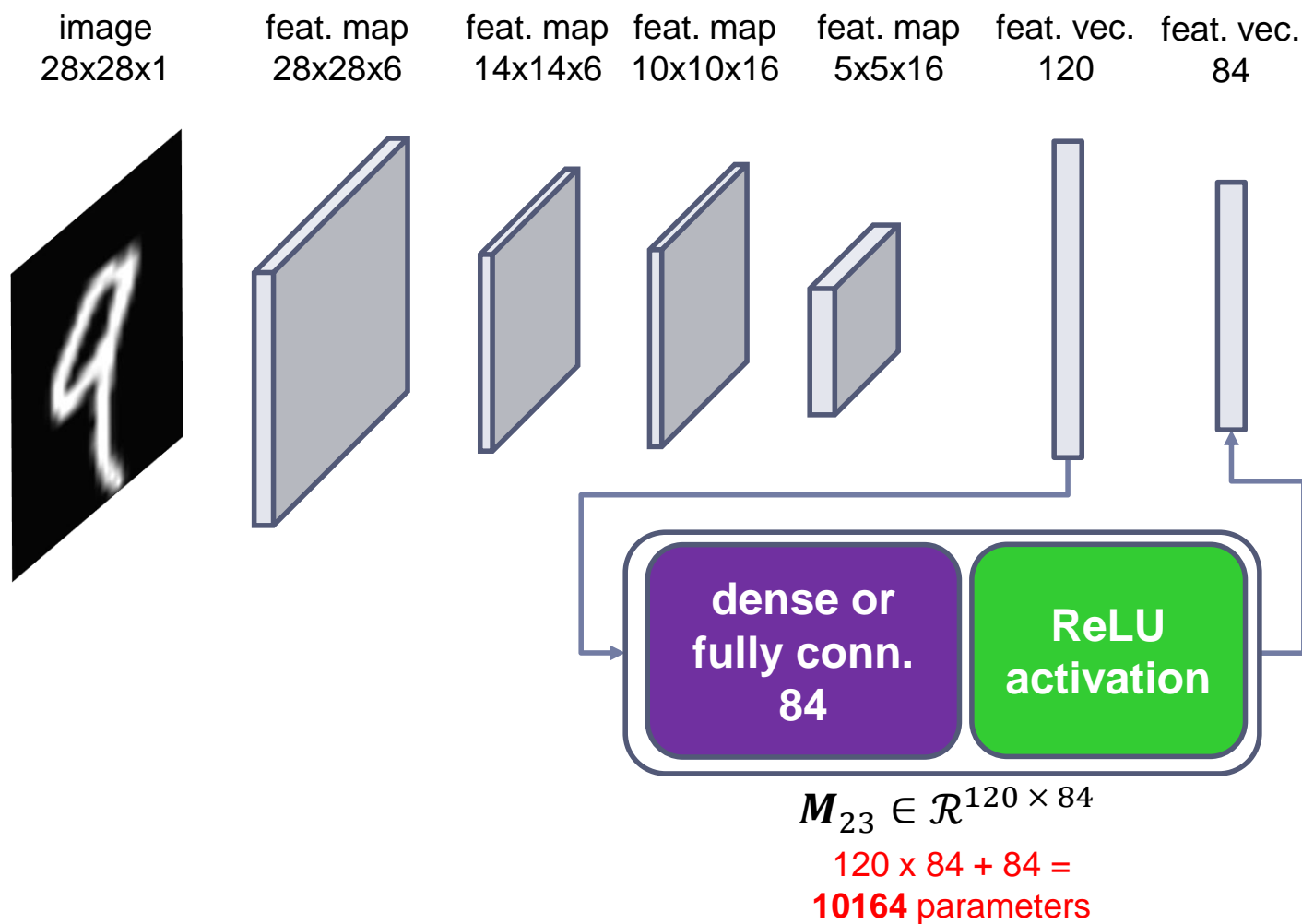
Architettura LeNet



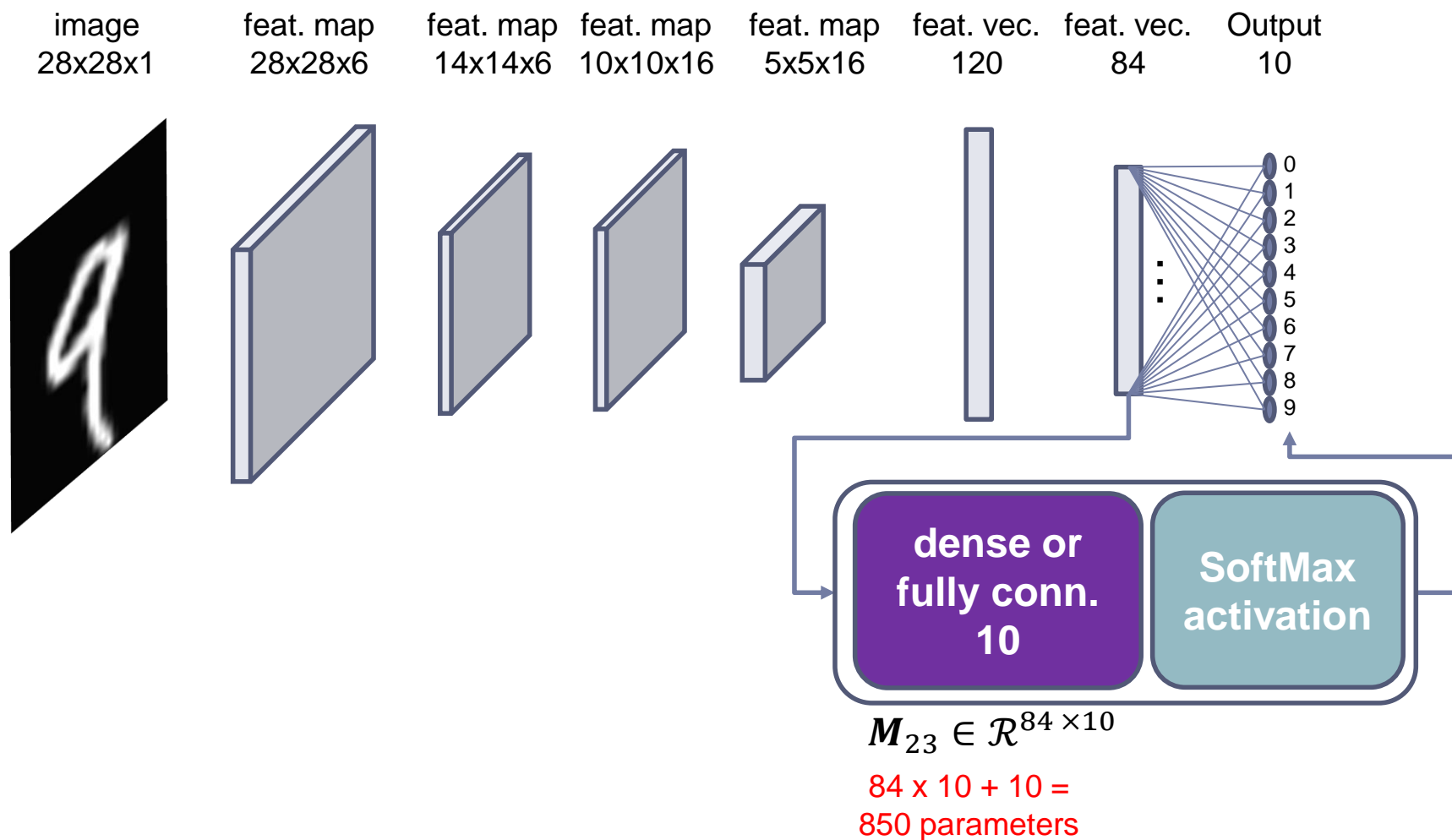
Architettura LeNet



Architettura LeNet



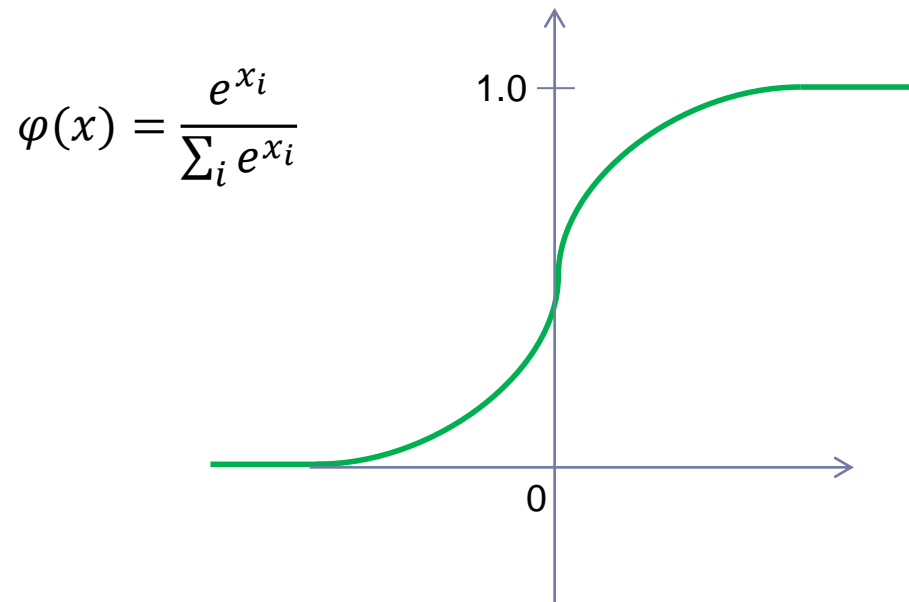
Architettura LeNet



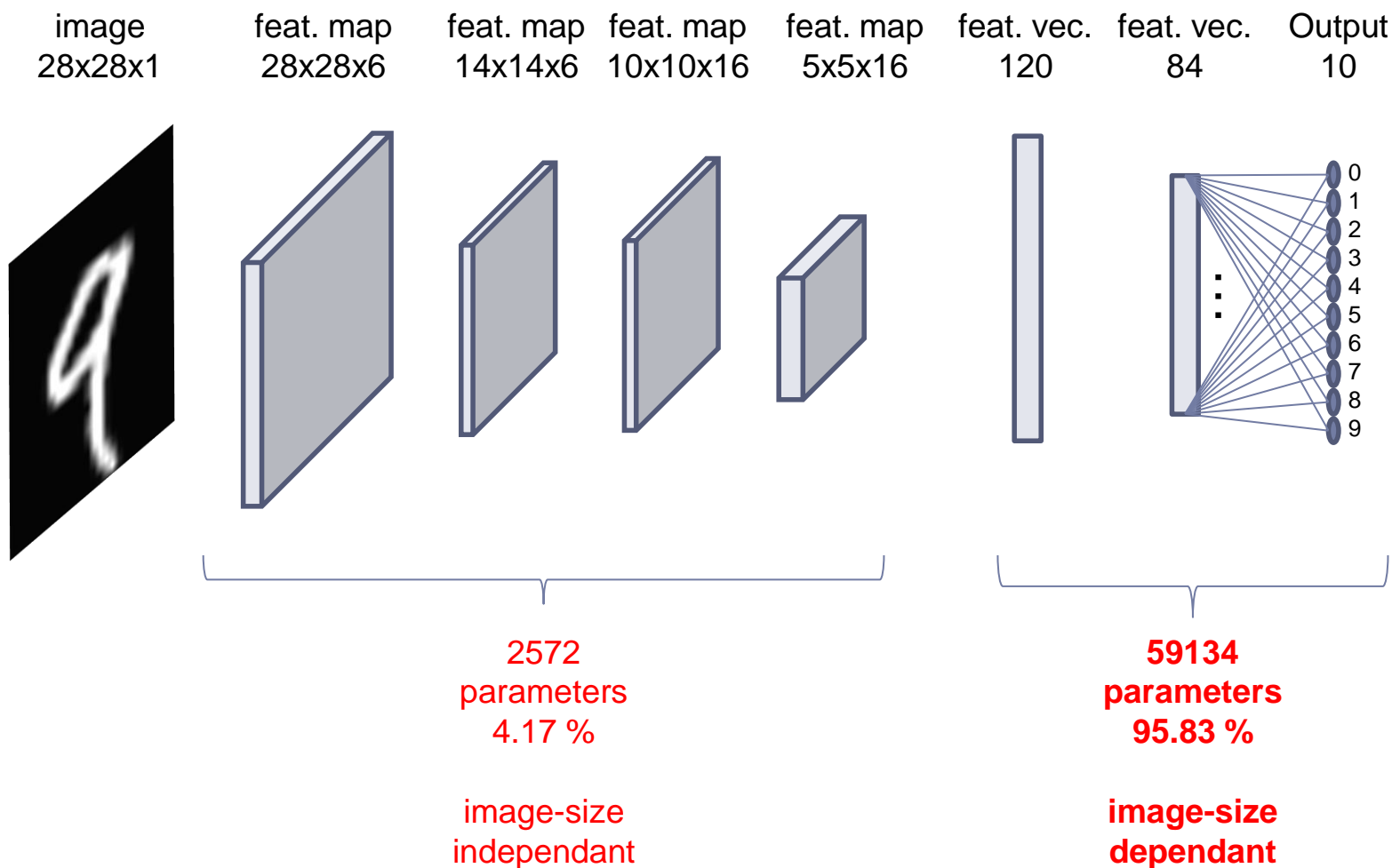
Softmax

L'output è nell'intervallo $[0, 1]$: si interpretare come una distribuzione di probabilità

Monotona: non cambia la classe predetta

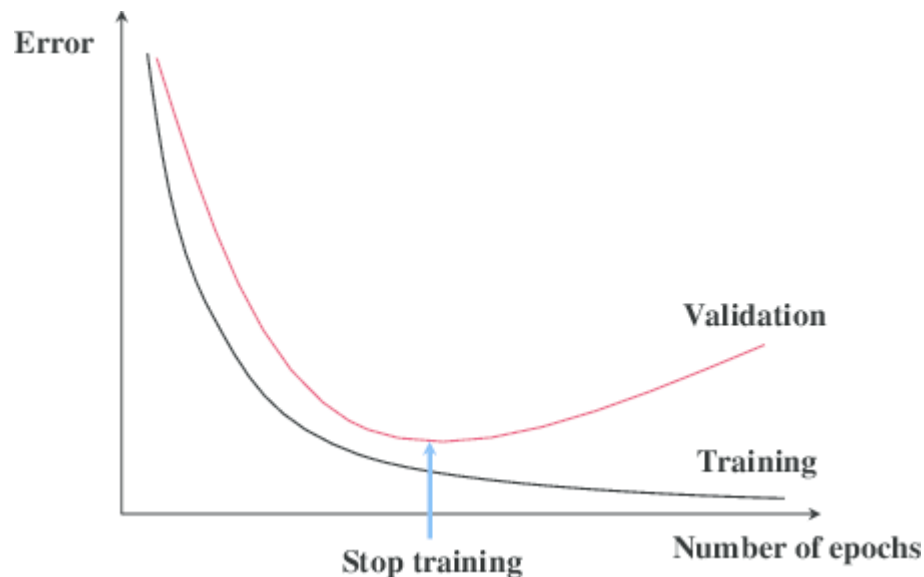


Architettura LeNet



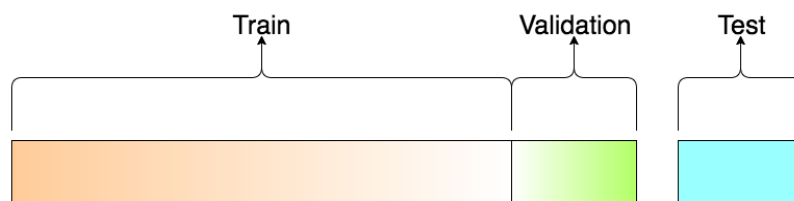
Training, validation, test

- Training set: I dati usati per addestrare la rete
- Validation set: I dati usati per settare I parametri ed evitare overfitting
- Test set: I dati usati per testare le prestazioni



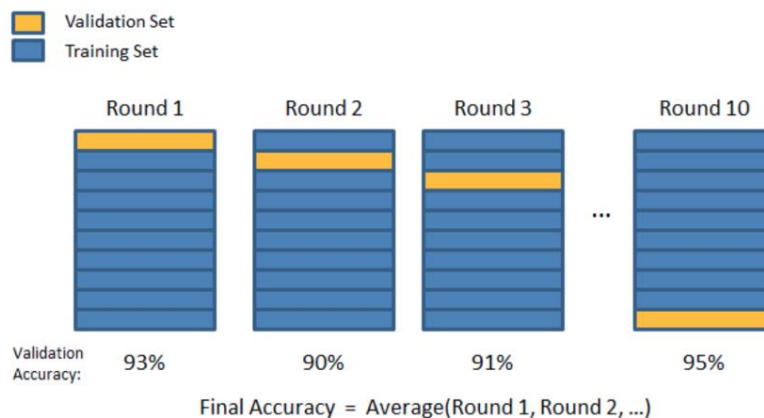
Dataset split

- Training (60%), validation (20%) e test (20%)



from: <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>

- Se il dataset è troppo piccolo: cross-validation



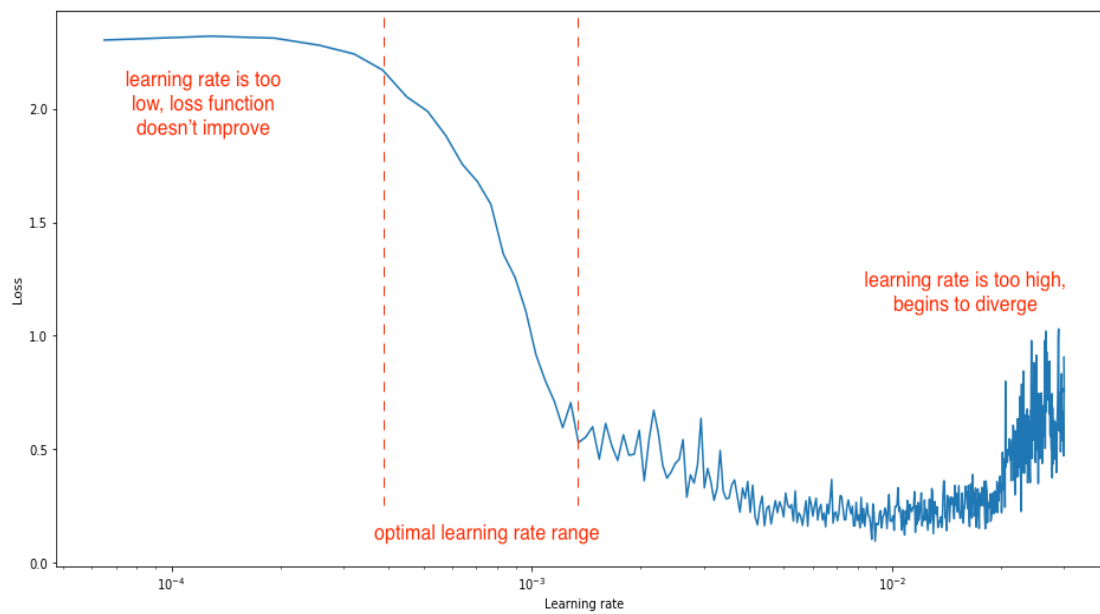
from: <https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6>

Dataset split

- Suddivisione per MNIST
 - Training set : 55000 immagini
 - Validation set : 5000 immagini
 - Test set : 10000 immagini

Come settare il learning rate

- Se la loss function decresce lentamente, bisogna aumentare il learning rate
- Se la loss function non decresce, diminuisci il learning rate
- Se la loss function oscilla, diminuisci il learning rate



from <https://www.jeremyjordan.me/nn-learning-rate/>

Come settare il learning rate

- Non è facile individuare il learning rate corretto 😊

