

Advanced Machine Learning

Terzo Assignment

Simone Paolo Mottadelli, 820786

Contents

1	Model architecture	3
2	Training	3
3	Testing	4

1 Model architecture

La Tabella 1 mostra l'architettura della CNN utilizzata per risolvere il task di classificazione sul dataset MNIST. L'ordine delle righe della tabella corrisponde all'ordine con cui i layer sono inseriti nella rete; dunque, si parte da un layer convoluzionale costituito da 20 filtri di dimensione $5 \times 5 \times 1$, fino ad arrivare ad un layer denso costituito da 10 neuroni, che costituiscono le 10 classi del dataset. In totale, la rete è costituita da 6995 parametri che devono essere appresi.

Input	Layer	Output	Numero di parametri
28x28x1	Convolution - $5 \times 5 \times 1 \times 20$	24x24x20	520
24x24x20	Activation ReLU	24x24x20	0
24x24x20	Max pooling - 2×2	12x12x20	0
12x12x20	Convolution - $3 \times 3 \times 20 \times 15$	10x10x15	2715
10x10x15	Activation ReLU	10x10x15	0
10x10x15	Max pooling - 2×2	5x5x15	0
5x5x15	Flatten	375x1	0
375x1	Dropout con rate 0.3	375x1	0
375x1	Dense - 10×1 con output function softmax	10x1	3760
		TOTAL:	6995

Table 1: Architettura della CNN

2 Training

Per allenare la CNN ho utilizzato la *categorical crossentropy* come loss function, perchè il problema è un problema di classificazione e l'output layer utilizza come funzione di output la softmax.

Ho deciso di utilizzare l'algoritmo di ottimizzazione *Adam* per allenare la CNN, poichè è un algoritmo molto performante e ampiamente utilizzato, utilizzando un *batch* di dimensione 128, 30 *epoche* e *learning rate* con valore di default 0.001.

Come tecnica di regolarizzazione, ho deciso di utilizzare *Dropout* nel livello denso della CNN, utilizzando come frazione di neuroni da disattivare un valore pari a 0.3.

Durante la fase di training, ho deciso di riservare il 10% del training set come validation set per monitorare le curve di apprendimento del modello. In particolare, la Figura 1 mostra sia l'accuracy ottenuta dal modello al variare delle epoche per il training set e per il validation set, sia il valore della loss function sempre al variare delle epoche sul training set e sul validation set. Il fatto che

le curve rispetto al validation set siano migliori delle curve rispetto al training set può essere imputato alla tecnica di regolarizzazione Dropout, in quanto, se nella fase di training ci sono dei neuroni che ad ogni iterazione sono disattivati, nella fase di predizione sul validation set la rete viene utilizzata nella sua interezza e nessun neurone è disattivato, quindi potenzialmente la rete è migliore e performa meglio.

Come possiamo vedere dalle immagini, all'ultima epoca l'accuracy sul train set e sul validation set sono circa 0.98 e 0.99, rispettivamente.

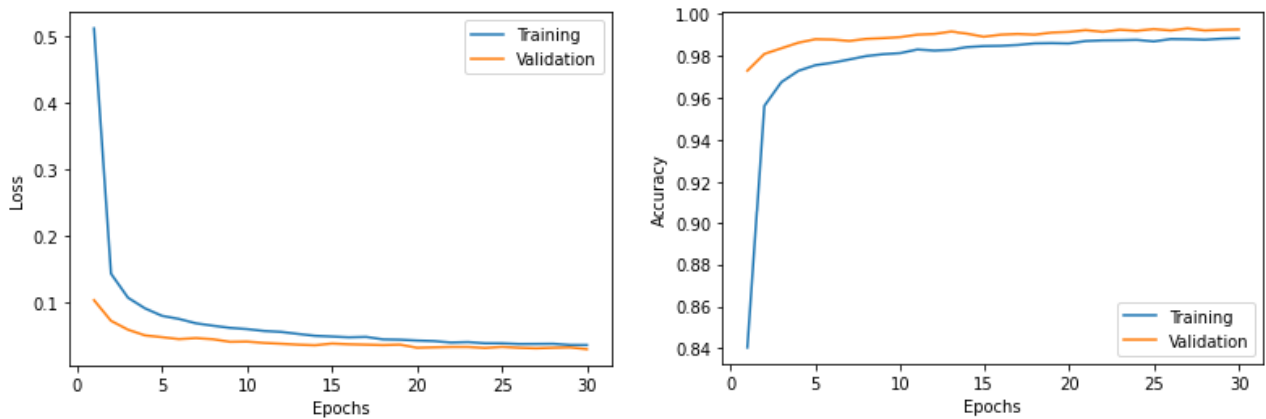


Figure 1: Loss function e accuracy durante la fase di training

3 Testing

Dopo aver trainato il modello, ho misurato le performance su un test set contenente circa il 14% del dataset MNIST. Ho ottenuto uno score di accuracy pari a 99.27%. La Figura 2 mostra la matrice di confusione ottenuta.

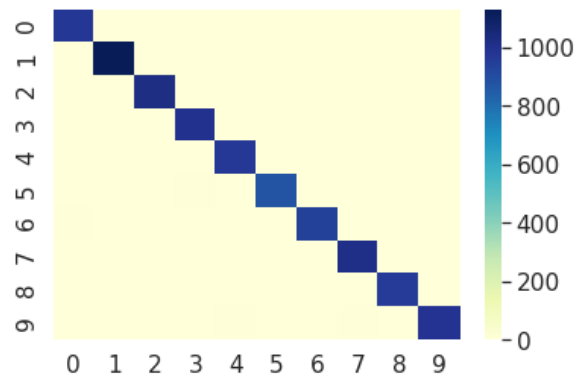


Figure 2: Matrice di confusione sul test set