

UNIVERSITÀ DEGLI STUDI DI  
MILANO-BICOCCA

ADVANCED MACHINE LEARNING

---

## Assignment 3

---

*Autore:*

Federico Manenti - 790032 - f.manenti3@campus.unimib.it

11 novembre 2019



# Indice

1	Dataset	1
2	Rete Neurale	1

## 1 Dataset

Il dataset scelto per questo task di classificazione è il *MNIST* uno dei più utilizzati in ambito accademico: consiste in immagini di cifre (da 0 a 9) in scala di grigi scritte a mano da diversi soggetti.

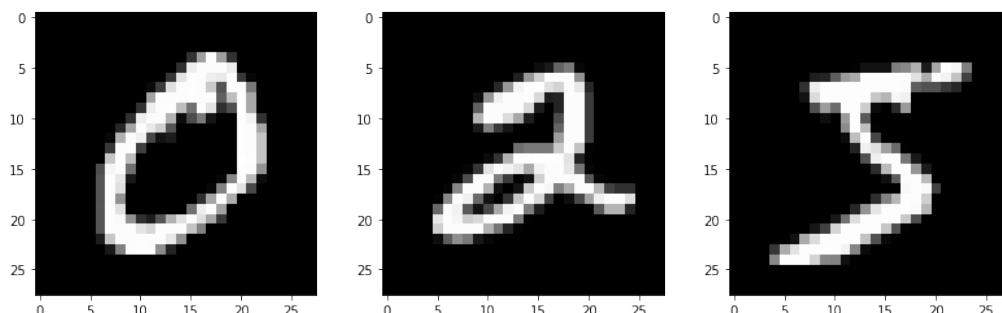


Figura 1: Tre esempi di cifre diverse

Le immagini utilizzate hanno una shape di (28,28,1) e i dati sono già divisi in train e test set.

## 2 Rete Neurale

Per risolvere il task assegnato è stata utilizzata una *CNN* con limite di 7500 parametri. Prima di utilizzare la rete neurale i dati hanno subito una *reshape* per avere dimensioni: ( $N^o$  immagini, 28, 28, 1). L'architettura utilizzata consiste in:

- **Conv2d** (16 neuroni, kernel = (3, 3), *Relu* ed essendo lo strato iniziale possiede l'input shape)
- **BatchNormalization** (parametri di default)

- **Conv2d** (16 neuroni, kernel = (3,3) e *Relu*)
- **BatchNormalization** (parametri di default)
- **MaxPoolig2d** (pool size = (2, 2))
- **Conv2d** (16 neuroni, kernel = (3,3) e *Relu*)
- **BatchNormalization** (parametri di default)
- **Conv2d** (8 neuroni, kernel = (3,3) e *Relu*)
- **BatchNormalization** (parametri di default)
- **MaxPoolig2d** (pool size = (2, 2))
- **Flatten**
- **Dropout** (rateo = 0.5)
- **Dense** (10 neuroni pari al numero di classi e *SoftMax* come funzione di attivazione)

La loss utilizzata è la *sparse categorical crossentropy*, l'ottimizzatore è *RM-Sprop* con learning rate di 0.0001 e il batch size è stato fissato a 64, inoltre è stato utilizzato il 30% di dati come validation set. Il numero di epoche è stato impostato a 100, ma per evitare overfitting si è implementato l'*early stopping* con pazienza di 5 e metrica da osservare la *validation accuracy*.

Il numero di parametri per ogni layer e in totale sono riportati nella figura seguente:

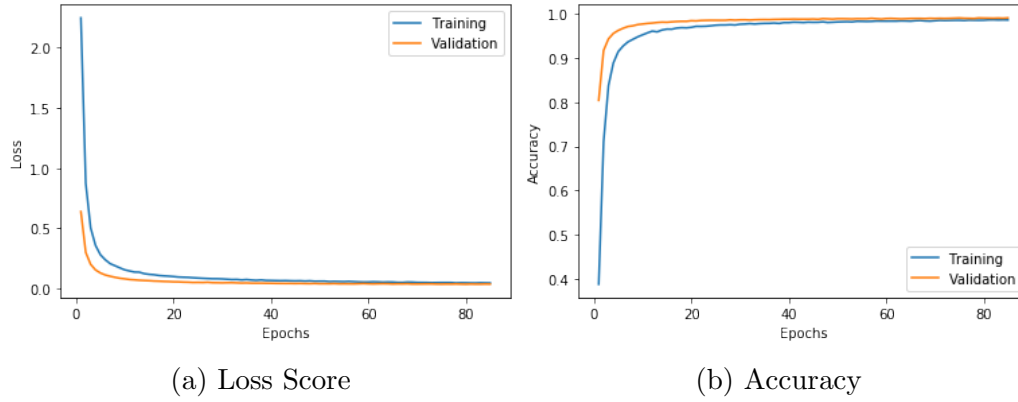
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 16)	160
batch_normalization_1 (Batch Normalization)	(None, 26, 26, 16)	64
conv2d_2 (Conv2D)	(None, 24, 24, 16)	2320
batch_normalization_2 (Batch Normalization)	(None, 24, 24, 16)	64
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 16)	0
conv2d_3 (Conv2D)	(None, 10, 10, 16)	2320
batch_normalization_3 (Batch Normalization)	(None, 10, 10, 16)	64
conv2d_4 (Conv2D)	(None, 8, 8, 8)	1160
batch_normalization_4 (Batch Normalization)	(None, 8, 8, 8)	32
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 8)	0
flatten_1 (Flatten)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290
Total params: 7,474		
Trainable params: 7,362		
Non-trainable params: 112		

Figura 2: Numero parametri

Grazie all'early stop il modello si ferma attorno all'ottantesima iterazione (nel caso considerato 85) e raggiunge i risultati:

- **Train Loss** = 0.0440
- **Train Validation** = 0.9856
- **Validation Loss** = 0.0345
- **Validation Accuracy** = 0.9903

I grafici relativi a Loss Score e Accuracy sono i seguenti:



Infine il modello è stato testato sulla porzione di dati apposita raggiungendo un risultato di **Loss** = 0.0268 e **Accuracy** = 0.9913.

Si riporta il classification report del test set:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	980
1	0.99	1.00	0.99	1135
2	0.99	0.99	0.99	1032
3	0.99	1.00	0.99	1010
4	0.99	0.98	0.99	982
5	0.99	0.99	0.99	892
6	0.99	0.98	0.99	958
7	0.98	0.99	0.99	1028
8	0.99	0.99	0.99	974
9	0.99	0.98	0.98	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000

Figura 4: Classification report