

UNIVERSITÀ DEGLI STUDI DI
MILANO-BICOCCA

ADVANCED MACHINE LEARNING

Assignment 4

Autore:

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

21 novembre 2019



Indice

1	Task e Dataset	1
2	Modello	2
2.1	Prima Prova	2
2.2	Seconda Prova	5
2.3	Terza Prova	6

1 Task e Dataset

Il task preso in considerazione consiste nella classificazione di immagini tramite un modello “classico” dopo aver tagliato la rete *VGG19*, a diverse profondità, pretrainata sul dataset *ImageNet*.

Il **Dataset** utilizzato è composto da 2180 immagini, in bianco e nero, delle cifre da 0 a 9 riprodotte con le mani, ogni figura ha dimensione 64 x 64.

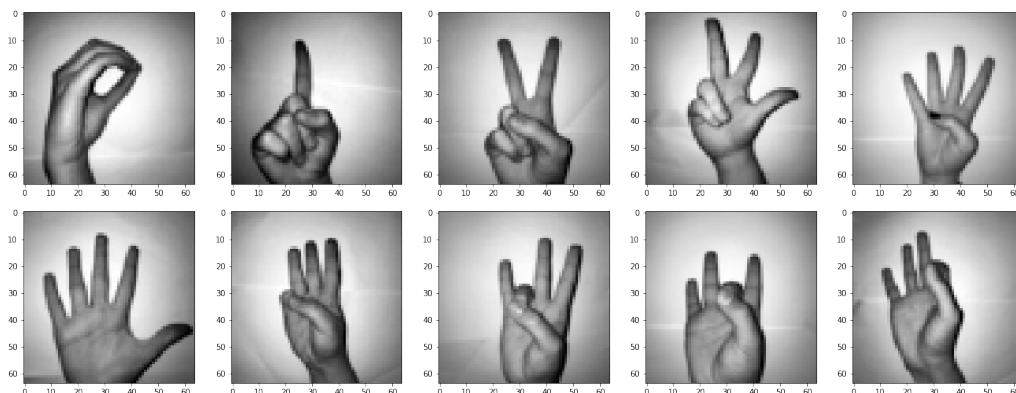


Figura 1: Esempio di un immagine per ogni label

Essendo le figure in bianco e nero sono state convertite in RGB grazie alla funzione *cvtColor* di *OpenCV* per poterle utilizzare con la *VGG19*. La distribuzione delle classi nel dataset, come si può osservare dalla seguente grafico, è abbastanza equilibrata e quindi non necessita di un resampling.

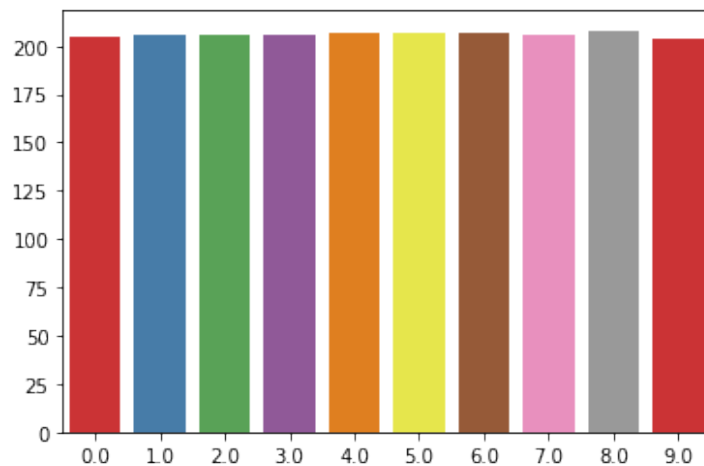


Figura 2: Distribuzione label immagini

Prima di poter passare i dati alla *CNN* sono stati reshaped ad una dimensione consona per l'utilizzo e divisi in train (85%) e test set (15%).

2 Modello

La *VGG19* è stata caricata con i pesi bloccati appresi grazie al dataset *ImageNet* e senza gli ultimi layer *Dense* utilizzati per la classificazione. Il classificatore “classico” utilizzato è una *SVM* di *sklearn* con kernel *rbf*, in ogni prova verrà utilizzata una tecnica di *GridSearch* per la ricerca degli iperparametri C e Γ migliori.

2.1 Prima Prova

Essendo il dataset utilizzato per il task abbastanza piccolo e diverso da *ImageNet* la *VGG19* è stata tagliata ad un livello abbastanza vicino all'inizio della rete: il layer “*block2_pool*” alla fine del secondo blocco. Per poter poi utilizzare le feature con un classificatore “classico” è stato inserito un layer *Flatten*.

L'architettura nella prima prova si presenta quindi così:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 64, 64, 3)	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
flatten_1 (Flatten)	(None, 32768)	0
Total params: 260,160		
Trainable params: 0		
Non-trainable params: 260,160		

Figura 3: Architettura prima prova

Dopo aver passato il dataset alla rete neurale ogni immagine è trasformata in una rappresentazione di 32768 feature utilizzate per addestrare e poi testare la *SVM*. Grazie alla *GridSearch* si trovano gli iperparametri migliori tra quelli pensati. In questo caso risultano **C**: 5 e **Gamma**: 0.1.

Sul train set il modello raggiunge performance pari a:

0.0	1.00	1.00	1.00	174
1.0	1.00	1.00	1.00	175
2.0	1.00	1.00	1.00	175
3.0	1.00	0.99	1.00	175
4.0	1.00	1.00	1.00	176
5.0	1.00	1.00	1.00	176
6.0	1.00	1.00	1.00	176
7.0	0.99	1.00	1.00	175
8.0	1.00	1.00	1.00	177
9.0	1.00	1.00	1.00	173
accuracy			1.00	1752
macro avg	1.00	1.00	1.00	1752
weighted avg	1.00	1.00	1.00	1752

Figura 4: Performance train set prima prova

Mentre i risultati sul test set sono mostrati nella seguente immagine:

	precision	recall	f1-score	support
0.0	1.00	0.94	0.97	31
1.0	1.00	0.97	0.98	31
2.0	0.91	0.97	0.94	31
3.0	1.00	1.00	1.00	31
4.0	0.87	0.84	0.85	31
5.0	1.00	1.00	1.00	31
6.0	0.90	0.90	0.90	31
7.0	0.86	0.81	0.83	31
8.0	0.88	0.97	0.92	31
9.0	0.94	0.97	0.95	31
accuracy			0.94	310
macro avg	0.94	0.94	0.94	310
weighted avg	0.94	0.94	0.94	310

Figura 5: Performance test set prima prova

2.2 Seconda Prova

Nella seconda prova la rete *VGG19* viene tagliata più in profondità: alla fine del terzo blocco, aggiungendo ancora un layer *flatten*. L'architettura quindi ha la seguente configurazione.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 64, 64, 3)	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv4 (Conv2D)	(None, 16, 16, 256)	590080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0
flatten_1 (Flatten)	(None, 16384)	0
Total params: 2,325,568		
Trainable params: 0		
Non-trainable params: 2,325,568		

Figura 6: Architettura seconda prova

In questo caso ogni immagine dopo essere passata nella rete neurale è trasformata in un vettore di feature di lunghezza 16384. Gli iperparametri usati per l'*SVM* sono **C**: 10 e **Gamma**: 0.015. I risultati raggiunti sul train set sono:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	174
1.0	1.00	1.00	1.00	175
2.0	1.00	0.99	1.00	175
3.0	0.99	0.99	0.99	175
4.0	1.00	0.99	1.00	176
5.0	1.00	1.00	1.00	176
6.0	0.99	1.00	1.00	176
7.0	0.99	1.00	1.00	175
8.0	1.00	1.00	1.00	177
9.0	1.00	1.00	1.00	173
accuracy			1.00	1752
macro avg	1.00	1.00	1.00	1752
weighted avg	1.00	1.00	1.00	1752

Figura 7: Performance train set seconda prova

Mentre sul test set il modello raggiunge le performance:

	precision	recall	f1-score	support
0.0	1.00	0.97	0.98	31
1.0	1.00	1.00	1.00	31
2.0	0.94	0.97	0.95	31
3.0	0.97	1.00	0.98	31
4.0	0.91	0.94	0.92	31
5.0	1.00	1.00	1.00	31
6.0	0.88	0.94	0.91	31
7.0	0.96	0.87	0.92	31
8.0	1.00	0.94	0.97	31
9.0	0.97	1.00	0.98	31
accuracy			0.96	310
macro avg	0.96	0.96	0.96	310
weighted avg	0.96	0.96	0.96	310

Figura 8: Performance test set seconda prova

2.3 Terza Prova

Come ultima prova la rete è stata tagliata alla fine del quarto blocco e come sempre è stato aggiunto un layer *flatten*.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 64, 64, 3)	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv4 (Conv2D)	(None, 16, 16, 256)	590080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0
block4_conv1 (Conv2D)	(None, 8, 8, 512)	1180160
block4_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv4 (Conv2D)	(None, 8, 8, 512)	2359808
block4_pool (MaxPooling2D)	(None, 4, 4, 512)	0
flatten_1 (Flatten)	(None, 8192)	0
Total params: 10,585,152		
Trainable params: 0		
Non-trainable params: 10,585,152		

Figura 9: Architettura terza prova

In questo caso le feature estratte sono di dimensione 8192, la *SVM* utilizza come iperparametri **C**: 20 e **Gamma**: 0.5 e le performance raggiunte sul train set sono presentate nella seguente immagine.

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	174
1.0	1.00	1.00	1.00	175
2.0	1.00	0.99	1.00	175
3.0	0.99	0.99	0.99	175
4.0	1.00	0.99	1.00	176
5.0	1.00	1.00	1.00	176
6.0	0.99	1.00	1.00	176
7.0	0.99	1.00	1.00	175
8.0	1.00	1.00	1.00	177
9.0	1.00	1.00	1.00	173
accuracy			1.00	1752
macro avg	1.00	1.00	1.00	1752
weighted avg	1.00	1.00	1.00	1752

Figura 10: Performance train set terza prova

Mentre sul test set:

	precision	recall	f1-score	support
0.0	1.00	0.97	0.98	31
1.0	0.97	1.00	0.98	31
2.0	0.90	0.90	0.90	31
3.0	0.94	1.00	0.97	31
4.0	0.87	0.87	0.87	31
5.0	1.00	1.00	1.00	31
6.0	0.85	0.90	0.88	31
7.0	0.86	0.81	0.83	31
8.0	0.97	0.90	0.93	31
9.0	0.94	0.94	0.94	31
accuracy			0.93	310
macro avg	0.93	0.93	0.93	310
weighted avg	0.93	0.93	0.93	310

Figura 11: Performance test set terza prova