

Assignment 4 (Advanced Machine Learning)

Davide Sangalli - Matricola: 848013

November 25, 2019

Abstract

In questo assignment si affronterà il problema del *Transfer Learning*, usando una rete neurale come *feature extractor* (effettuando tagli a diversi livelli della rete), per poi utilizzare tali *features* come input per un classificatore "tradizionale", in particolare si è scelta la *Logistic Regression*.

1 Scelta del dataset e della rete neurale pre-trainata

1.1 Architettura della rete neurale

La rete neurale scelta è stata la *VGG16*, pre-trainata sull' *ImageNet*, ovvero un dataset contenente oltre 14 milioni di immagini appartenenti a 1000 categorie diverse.

Tale rete neurale ha la seguente struttura:

- un *input_layer* con dimensione delle immagini (224,224,3), ovvero delle immagini dimensione 224x224 pixels e a colori (3 canali di colore, RGB)
- 5 blocchi convoluzionali, ognuno dei quali costituito da 3 layer convoluzionali e un layer di *MaxPooling*
- 2 layers *fully-connected*
- un layer finale di predizione

1.2 Scelta del dataset

Il dataset scelto è composto da 10 classi, ovvero immagini di cani, cavalli, elefanti, farfalle, galline, gatti, mucche, pecore, ragni e scoiattoli.

Il dataset è stato così suddiviso:

- per il training set sono state usate 300 immagini di ogni classe, per un totale di 3000 immagini

- per il training set sono state usate 200 immagini per ogni classe, per un totale di 2000 immagini

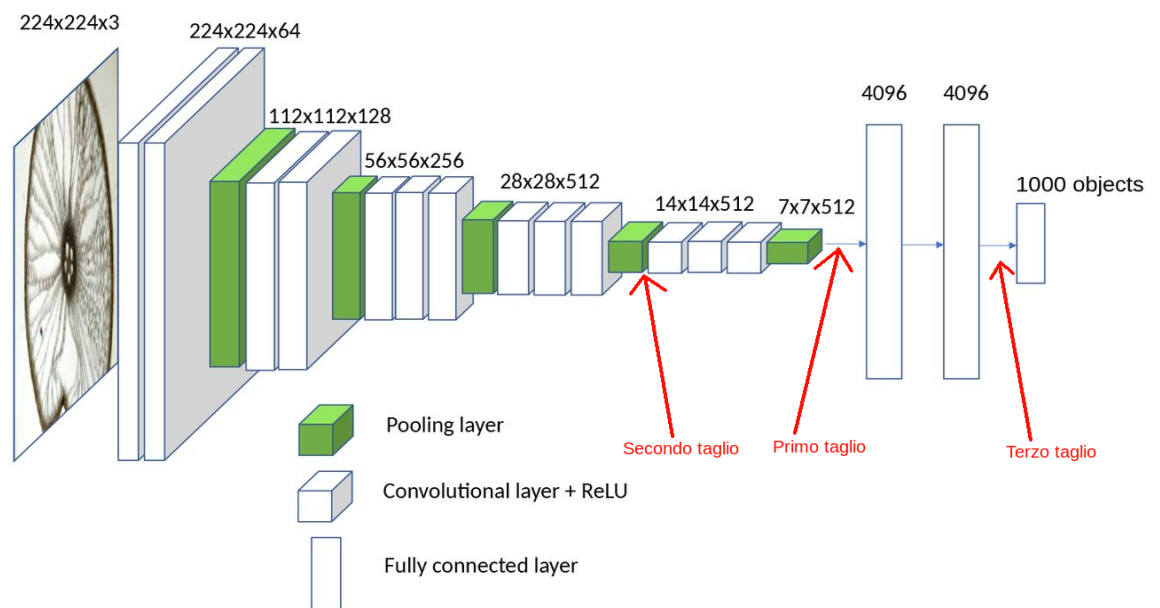
Non è stato utilizzato tutto il dataset, poichè conteneva troppe immagini e questo avrebbe reso molto oneroso il caricamento dei dati in memoria.

E' stato perciò utilizzato un *random sampling*, facendo attenzione a non usare una stessa immagine sia per training set che per il test set.

2 Classificazione tramite la *Logistic Regression*

2.1 Schema dei tagli

Viene riportato di seguito lo schema dei tagli effettuati sulla *VGG16*. Nei punti successivi verranno analizzati nel dettaglio.



2.2 Primo taglio

Il primo taglio alla rete neurale è stato fatto a livello del quinto blocco convoluzionale (comprendendo anche il layer di *MaxPooling*), che ha una shape di $(7,7,512)$. Le *features* estratte sono perciò pari a $7*7*512=25088$.

Ovviamente, per poterle dare in input alla *Logistic Regression*, è stato prima eseguito un *flatten* su di esse.

L'accuracy di questo primo modello di classificazione è del 92%, come confermato dal *classification_report*, riportato di seguito. Viene inoltre riportata anche la *confusion_matrix*.

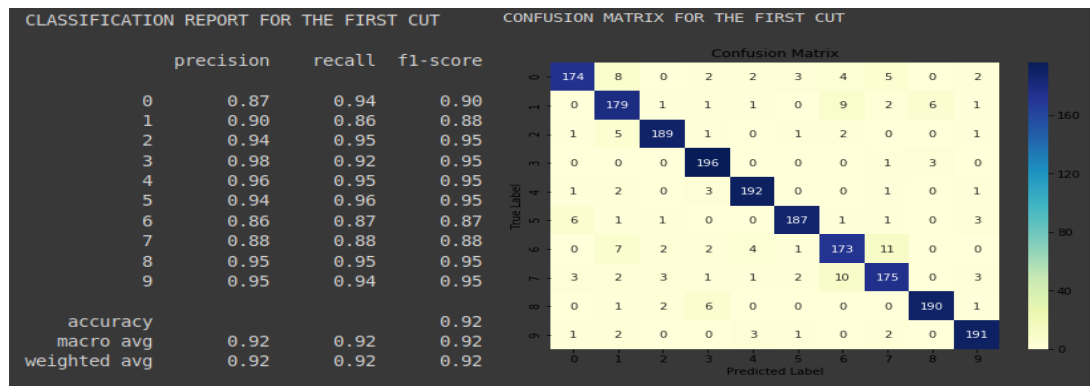


Figure 1: Classification report relativo al test set sul primo taglio

Figure 2: Confusion matrix relativa al test set sul primo taglio

2.3 Secondo taglio

Il secondo taglio alla rete neurale è stato fatto a livello del quarto blocco convoluzionale (comprendendo anche il layer di *MaxPooling*), che ha una shape di $(14, 14, 512)$.

Le *features* estratte sono perciò pari a $14 * 14 * 512 = 100352$, dopo aver eseguito un *flatten* su di esse.

L'accuracy di questo secondo modello di classificazione è dell' 82%. Anche in questo caso vengono riportati sia il *classification_report* che la *confusion_matrix*.

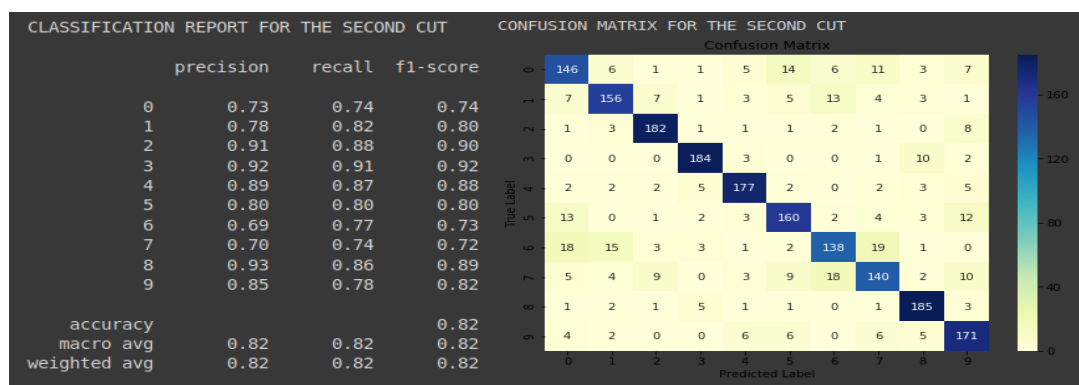


Figure 3: Classification report relativo al test set sul secondo taglio

Figure 4: Confusion matrix relativa al test set sul secondo taglio

2.4 Terzo taglio

Inizialmente è stato provato un taglio più in alto, a livello del terzo blocco, ma l'accuracy era bassa (intorno al 66%) ed inoltre il tempo richiesto per il training del modello era molto elevato, perciò si è optato di tagliare la rete ad un altro livello.

E' stato deciso di utilizzare tutta la rete (compresi i layers *fully_connected*), tagliandone solamente l'ultimo layer (ovvero quello di classificazione).

In quest'ultimo caso, le features estratte sono perciò pari a 4096.

L'accuracy di questo modello è pari al 93%, come mostrato di seguito nel *classification_report*. Viene mostrata anche la *confusion_matrix*.

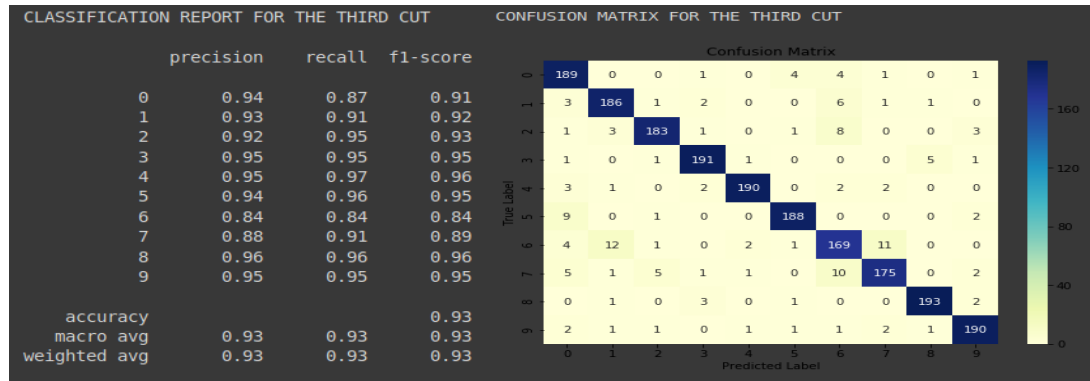


Figure 5: Classification report relativo al test set sul terzo taglio

Figure 6: Confusion matrix relativa al test set sul terzo taglio

3 Note

3.1 Note sull'uso della *Logistic Regression*

E' stata usata una *Logistic Regression* con i seguenti parametri:

- *class_weight*="balanced"
- *multi_class*="auto"
- un numero massimo di iterazioni pari a 200 (*max_iter*=200)
- *random_state*=1, per fissare un seme e rendere i risultati riproducibili
- *solver*="newton-cg", poichè particolarmente indicato per una classificazione multiclasse

In fase di classificazione è stata utilizzata anche una *SVM* con kernel lineare. Anche questo classificatore funzionava bene (i risultati dell'accuratezza erano del tutto simili a quelli ottenuti dalla regressione logistica), ma durante la fase di training era molto più lento della *Logistic Regression*, perciò si è optato di scegliere quest'ultima come classificatore.

3.2 Note sulla *VGG16*

Per tutti e tre i tagli effettuati si verifica *overfitting*, poichè tutti i *training_set* hanno un'accuracy pari al 100%, mentre tutti i *test_set* hanno un'accuracy minore (circa 92%-93% al massimo).

Questo fenomeno però è giustificato dal fatto che la *VGG16* è stata allenata utilizzando il dataset *ImageNet*, contenente oltre 14 milioni di immagini, appartenenti a 1000 classi differenti, mentre il *training_set* utilizzato per questo problema è composto solo da 3000 immagini, appartenenti a 10 classi differenti. E' perciò normale l'*overfitting* che si presenta.

4 Conclusioni

Come si nota dai *classification_reports*, più il taglio è in alto, più l'accuracy diminuisce e viceversa. Infatti, la migliore accuratezza si raggiunge con l'ultimo taglio, dove sono stati tenuti i layers *fully_connected* della rete neurale (93% di accuracy).

La classe che in tutti e tre i casi è stata classificata peggio è stata quella della "mucca".

Inoltre, più il taglio della rete avviene in basso, meno è il tempo richiesto dal classificatore per la computazione e viceversa.