Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Object category recognition

Authors:
Fabbris Elia - 793240 - e.fabbris1@campus.unimib.it
Fumagalli Matteo - 793670 - m.fumagalli85@campus.unimib.it

July 8, 2019

Abstract

Nella seguente relazione viene descritto come sono state addestrate delle reti neurali convoluzionali per classificare delle immagini realistiche. Questo è uno dei punti facenti parte di una sfida chiamata 'The VOC2012 Challenge', ossia essere in grado di predirre la presenza di una o più classi all'interno di un'immagine di test. Le classi presenti nel dataset sono venti e si riferiscono a persone, animali ed oggetti reali. Sono state addestrate tre reti: una multi-class, una multi-label ed una multi-task mutli-label. In tutte è stato usato il *transfer-learning* di reti preaddestrate.

1 Introduzione

Lo scopo principale è stato quello di creare una o più reti ad apprendimento supervisionato che fossero in grado di predirre con una buona accuratezza la presenza in una data immagine di determinate classi. Questo è il primo punto di una sfida chiamata 'The VOC2012 Challenge' conclusa a fine 2012 in cui la miglior rete ha avuto un'accuratezza media globale del 63.48%. Dal momento che però non sono state rese disponibili le classi associate a ciascuna immagine di test non è stato possibile utilizzare il dataset di test fornito dalla competizione e quindi avere un paragone valido. Per la fase di test è stato deciso quindi di utilizzare una porzione delle immagini appartenenti al dataset di addestramento.
Per la risoluzione del compito è stato deciso di procedere in tre metodi differenti. Un primo approccio è basato su una rete convoluzionale multi-class (CNN Multi-class), in cui ciascuna immagine di test viene associata ad una singola classe. Un secondo approccio è basato invece su una rete convoluzionale multi-label (CNN Multi-label) ed un terzo su di una multi-task multi-label (MT-CNN Multi-label); questi ultimi permettono di associare ad un'immagine, a differenza della prima, una o più classi.
Per motivi riguardanti il dataset, i quali saranno esposti più nel dettaglio nel Capitolo 2, è stato necessario utilizzare dei modelli di reti convoluzionali preaddestrati su dataset di dimensioni maggiori. Tutto ciò dato dal fatto che utilizzare uno di quei modelli avrebbe permesso di ottenere una maggiore generalizzazione. Le reti prese in considerazione sono state: 'VGG16'[1] per il primo approccio, 'Inception-V3' [2] per il secondo e 'VGG19' [1] per il terzo. Nel Capitolo 3 verranno definite nel dettaglio le operazione di *transfer-learning* eseguite e verranno descritte le strutture finali delle reti.

Nel Capitolo 4 saranno esposti i risultati di ciascuna rete e messi a confronto e nel Capitolo 5 verranno discussi.

2 Dataset

In questo elaborato è stato utilizzato il dataset fornito per la competizione 'The VOC2012 Challenge'. Le classi di cui è composto sono le seguenti:

- Persone: persona;
- Animali: uccello, gatto, cane, cavallo, pecora e mucca;
- Veicoli: aereo, bicicletta, barca, bus, motocicletta, macchina e treno;
- Indoor: bottiglia, sedia, tavolo da pranzo, piante in vaso, divano e monitor TV.

Inizialmente è stato creato un file '.csv' di supporto per trovare le label assegnate ad ogni immagine in modo da poter essere in grado di manipolare il dataset iniziale.
I dataset che sono stati utilizzati per la fase di addestramento e validazione delle reti sono due. Uno composto da immagini alle quali è stata associata una singola label ed un secondo invece composto sia da immagini con label singole che multiple.
Nel primo dataset è stato effettuato un bilanciamento per ogni classe. Successivamente è stato diviso in 80% train, 10% validation, 10% test ed è stato utilizzato nell'approccio multi-class.
Il secondo dataset è stato utilizzato per intero sia nell'approccio multi-label sia multi-task: nel primo diviso in 50% train, 40% validation e 10% test, nel secondo diviso 50% train, 40% validation e 10% test.
In Figura 1 è riportato un grafico che illustra la numerosità di ciascun dataset utilizzato.

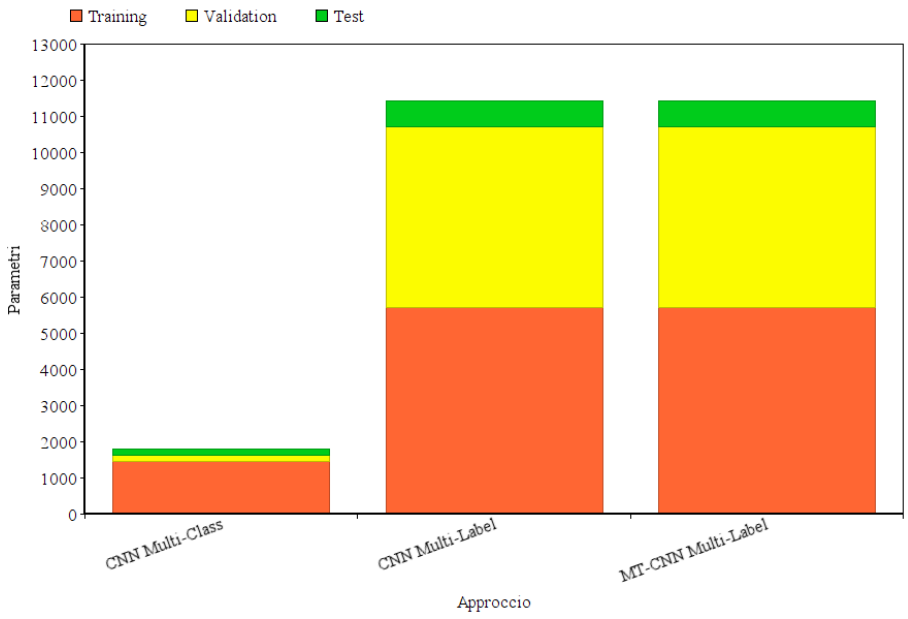

Figure 1: Dimensione dei dataset utilizzati.

3 Approcci metodologici

L'architettura relativa alle reti che sono state realizzate viene descritta nei paragrafi seguenti. Ciascuna rete, come riportato precedentemente, è stata realizzata tramite *transfer-learning* precisamente sulle reti VGG16 [1], VGG19 [1] ed InceptionV3 [2] preaddestrate su dataset *ImageNet*. Le principali ragioni per cui è stato scelto di effettuare quest'ultimo passaggio sono le seguenti:

- garantire una maggiore generalizzazione dei dati data la numerosità del dataset a disposizione (*ImageNet*: $>$ 14mln immagini, $>$ 20k categorie; dataset utilizzato: $>$ 10k, 20 immagini, 20 categorie);
- forte somiglianza tra le feature estratte dai due dataset.

3.1 CNN Multi-class

Per questo approccio sono state utilizzate le immagini single-label.
Inizialmente era stato deciso di procedere con una definizione di una rete custom, creata apposta per questo dataset; siccome la numerosità del dataset

non risulta essere sufficiente, è stato deciso di utilizzare la rete 'VGG16'[1] tramite *transfer-learning*, preaddestrata sul dataset *ImageNet*, alla quale sono stati eliminati gli ultimi 4 strati.
Gli strati restanti sono stati mantenuti con i pesi bloccati e successivamente sono stati inseriti oltre allo strato *Flatten()*, uno strato *Dense(256, activation='relu')*, uno strato *Dropout(0.5)* e uno strato di output *Dense(20, activation='softmax')*.
Le immagini in input sono state impostate con *shape = (224,224,3)* e la *batch size* è stata settata pari a 16.

Come loss function è stata scelta la *categorical_crossentropy*, dato che come output, su una certa immagine di input, si vuole la probabilità per ogni classe presente nell'elaborato.
Come optimizer è stato scelto *RMSprop* con *learning rate* pari a *1e-4*. Confrontandolo con l'optimizer *Adam*, sono state riscontrate delle performance migliori e quindi la scelta è ricaduta sull'optimizer *RMSprop*.
Come funzione di attivazione è stata utilizzata la *relu* per tutti gli strati, tranne che per l'ultimo in cui è stata utilizzata la *softmax*; questo perchè restituisce in output la probabilità per ogni target class e quindi adatta al compito.
Le metrica presa in considerazione è stata l'*accuracy* .

3.2 CNN Multi-label

Per lo sviluppo di una rete multi-label, dopo varie prove, è stato scelto di effettuare *transfer-learning* su 'InceptionV3' [2]. Essa a differenza di altre reti come le VGG estrae le feature su più livelli, ovvero in uno stesso modulo utilizza filtri di differenti misure (1x1, 3x3 e 5x5) in modo da catturare dettagli su più scale, dopodiché unisce i risultati in un solo output e lo pone in input al modulo successivo.
Alla rete è stato rimosso l'ultimo strato (output) e sono stati bloccati tutti i pesi appartenenti agli altri strati. Dopodiché sono stati inseriti i seguenti strati: *Dense(512, activation = 'relu')*, *Dropout(0.3)* e *Dense(20, activation='sigmoid')*.
Come funzione di attivazione dello strato di output è stata utilizzata la sigmoide poiché, a differenza di una rete multi-class, una rete multi-label deve restituire in uscita la probabilità che una classe sia presente nell'immagine indipendentemente dalla presenza di un'altra classe. La soglia che è stata fissata per stabilire se una classe sia presente o meno in una immagine è stata

fissata a 0.5. Sempre per questo motivo è stato necessario stabilire come funzione di loss *binary_crossentropy*.
La funzione di ottimizzazione che è stata impiegata è *Adam* ed è stato utilizzato un fattore di apprendimento *lr* pari a *1e-3*. La metrica che è stata monitorata durante la fase di addestramento è l'*accuracy*.

3.3 MT-CNN Multi-label

Per lo sviluppo di una rete convoluzionale multi-task è stato scelto di effettuare *transfer-learning* su 'VGG19'. Dopo varie prove sull'architettura che avrebbe dovuto avere la rete e monitorando i vari risultati è stato scelto che la struttura condivisa migliore fosse la seguente: 'VGG19' (strato finale di output rimosso e pesi bloccati dei restanti strati) e strato *Dense(512, activation = 'relu')*. Dopodiché la rete è stata divisa in venti task, ciascuno corrispondente ad una differente classe. Ogni sotto struttura è stata composta dai seguenti strati: *Dense(128, activation = 'relu')* e output *Dense(1, activation='sigmoid')*.
Come funzione di attivazione per ciascun strato di output, dato che ognuno è stato composto da un singolo neurone, è stata utilizzata la sigmoide. Per lo stesso precedente motivo come funzione di loss è stata impiegata *binary_crossentropy*. La funzione di ottimizzazione impiegata è *Adam* con un fattore di apprendimento *lr* pari a *1e-3*. Come metrica per ogni singolo task è stata presa in considerazione l'*accuracy*.

4 Risultati e valutazioni

Nei paragrafi seguenti vengono mostrati i risultati ottenuti nei vari approcci. Oltre ai risultati vengono mostrate le impostazioni e i parametri scelti per la parte di train.

Table 1: Impostazioni dei vari approcci.

	Epoche	Callbacks	Funz. Attiv.	lea. rate
CNN Multi-class	5	EarlyStopping	RMSProp	0.0001
CNN Multi-label	1	–	Adam	0.001
MT-CNN Multi-label	1	–	Adam	0.001

4.1 CNN Multi-class

Le metriche *accuracy* e *loss* monitorate durante la fase di addestramento hanno restituito i seguenti risultati:

- *Train*: *accuracy*: 67%, *loss*: 1.01;
- *Validation*: *accuracy*: 53%, *loss*: 1.40;
- *Test*: *accuracy*: 53,1%.

4.2 CNN Multi-label

Le metriche *accuracy* e *loss* monitorate durante la fase di addestramento hanno raggiunto i seguenti risultati:

- *Train*: *accuracy*: 95%, *loss*: 0.15;
- *Validation*: *accuracy*: 96%, *loss*: 0.09.

Un'altra metrica che è stata misurata negli approcci multi-label è la *Hamming-loss*. Essa si basa sulla differenza tra due stringhe di bit rappresentanti le label originali e predette, pone in XOR bit a bit e calcola la media sul tutto il dataset. La formula è la seguente:

$$HammingLoss = \frac{1}{|N|\,|L|} \sum_{i=1}^{N} \sum_{j=1}^{L} XOR(y_{i,j}, \hat{y}_{i,j})$$

Nella fase di test sono stati ottenuti i seguenti risultati:

- *accuracy*: 52.4%;
- *Hamming-loss*: 0.04.

4.3 MT-CNN Multi-label

Infine nella rete multi-task, ottenuta tramite *transfer-learning* sulla rete 'VGG19', sono stati ottenuti nella fase di test i seguenti risultati:

- *accuracy*: 43.5%;
- *Hamming-loss*: 0.04.

Le metriche *accuracy* e *loss* monitorate durante la fase di addestramento hanno raggiunto i seguenti risultati:

- *Train*: *accuracy*: 94%, *loss*: 0.03;
- *Validation*: *accuracy*: 96%, *loss*: 0.03.

Nella Tabella 2 vengono esposti i risultati che sono stati ottenuti nella fase di test nei tre differenti approcci sviluppati.

Table 2: Risultati ottenuti.

	Test accuracy
CNN Multi-class	53.1%
CNN Multi-label	52.4%
MT-CNN Multi-label	43,5%

5 Discussioni

Dai risultati ottenuti si può vedere come delle reti neurali, in questo caso preaddestrate, siano state in grado, dato un dataset relativamente piccolo, di ottenere una buona accuratezza sui test. Presumibilmente per ottenere dei migliori risultati sarebbe stata necessaria maggiore capacità computazionale e un dataset di dimensioni maggiori in modo da poter utilizzare o reti di dimensioni maggiori o reti custom addestrate su questo preciso compito.
Di seguito verranno esposti per ogni approccio le varie problematiche ed interpretazione dei risultati.

5.1 CNN Multi-class

L'idea iniziale, ossia quella di creare una rete custom da addestrare, è stata scartata data la poca numerosità del dataset.
La scelta di basare gli esperimenti su reti già addestrate, in questo caso 'VGG16', è risultata valida. Pur avendo un dataset piccolo sono stati ottenuti risultati soddisfacenti. Attraverso l'utilizzo dell'*early stopping* è stato limitato l'*overfitting*: è stato applicato in maniera abbastanza aggressiva evitando così la crescita del valore di *loss* pur perdendo in termini di *accuracy*.

5.2 CNN Multi-label

Utilizzare la rete preaddestrata 'InceptionV3' si è dimostrata essere una efficace strategia, ottenendo buoni risultati sul dataset di test impiegato. L'accuratezza ottenuta utilizzando circa 700 immagini di test sia multi-label che single-label è stata del 52,4%. Anche se non è stato possibile utilizzare come paragone il risultato ottenuto dalla miglior rete della competizione (63,5%), quello ottenuto è stato reputato un risultato soddisfacente. E' stato effettuato anche *transfer-learning* su rete 'VGG19' ottenendo però dei risultati decisamente peggiori. Si è presupposto che nell'approccio multi-label sia più efficace la rete 'InceptionV3' data la miglior capacità di estrarre feature.

5.3 MT-CNN Multi-label

Anche se con la seguente rete si è ottenuto un risultato peggiore, sul medesimo dataset di test, rispetto alla 'CNN Multi-label' è stato considerato comunque soddisfacente. In una prima prova è stata addestrata la medesima rete solamente con immagini single-label. Tale metodo si è dimostrato poi efficace sì su immagini single-label ma non su quelle multi-label.
Successivamente si è presupposto che il numero di immagini di addestramento composte da più label sia minimo e che quindi un dataset avente in particolare un maggior numero di queste ultime avrebbe portato quasi per certo ad un miglioramento sia della seguente rete che della rete precedente.

6 Conclusioni

Prendendo spunto dalle discussioni del capitolo precedente è stato possibile affermare che l'approccio 'CNN Multi-class' è stato effettuato più per prova che per effettiva necessità dato che il dataset si presta meglio ad approcci multi-label. Il rischio di *overfitting* è alto e se si volesse procedere con questo tipo di approccio bisognerebbe aumentare il dataset con i label singoli.
Passando invece agli approcci tramite reti multi-label supervisionati è stato possibile capire che, pur con risorse limitate, è stato possibile raggiungere performance dignitose. Infatti anche se pur non potendo fare un confronto diretto coi risultati ottenuti dalle reti che parteciparono alla competizione, sia per il fatto che le reti preaddestrare utilizzate nella seguente prova non erano ancora presenti al tempo e il dataset di test sia differente, i risultati ottenuti da entrambe le reti multi-label sono stati reputati soddisfacenti.
Come espresso precedentemente risultati migliori si sarebbero potuti ottenere con un dataset, in particolare riguardo alle immagini multi-label, di dimensioni maggiori.
Per eventuali sviluppi futuri si potrebbe lavorare e migliorare gli approcci multi-label, in particolare l'approccio multi-task, utilizzando sempre la stessa rete di base e aumentando la numerosità delle immagini.

References

[1] S. I. J. S. Z. W. Christian Szegedy, Vincent Vanhoucke, "Rethinking the Inception Architecture for Computer Vision," 2015.

[2] A. Z. Karen Simonyan, "Very deep convolutional networks for large-scale image recognition," 2016.