

Image classification basato su reti neurali coinvoluzionali

Lorenzo Meroni

Univerisita' degli studi Milano Bicocca, Milano
l.meroni16@campus.unimib.it

Abstract. Intel image classification <https://www.kaggle.com/datasets/puneet6060/intel-image-classification> è un dataset presente sulla piattaforma Kaggle che raccoglie immagini in formato *jpg* appartenenti a 6 differenti categorie.

Lo scopo dello studio è prevedere, in base a specifici modelli basati sulle reti neurali, allenati su un training set a che categoria di appartengano le immagini presenti nel test set.

Il miglior risultato è stato ottenuto mediante una rete neurale coinvoluzionale (cnn) minimizzando una funzione di perdita e massimizzando l'accuracy.

Keywords: Intel image classification · previsione · reti neurali coinvoluzionali

1 Introduzione

Il problema preso esame è un problema di classificazione di diverse categorie di immagini nel test set utilizzando come mezzi modelli basati sulle reti neurali, cuore centrale del corso.

Il modello è stato allenato usando una parte del training set mentre la restante parte è stata utilizzata per validare i risultati. Il modello che è stato scelto è una rete neurale coinvoluzionale con uno strato di input, 3 strati di coinvoluzione seguiti rispettivamente ognuno da un layer di pooling che ha il compito di eseguire un'aggregazione delle informazioni, generando così feature map di dimensione inferiore, e un *layer dropout* per evitare l'overfitting infine uno strato fully connected.

L'ultimo strato è formato da una funzione di attivazione di tipo softmax perché l'output è multiclasse.

Il risultato ottenuto è un'accuracy sul validation set dello 0.8.

2 Materiali e metodi

2.1 Materiali

Il dataset preso in esame è un dataset di immagini in formato *jpg* presente sulla piattaforma Kaggle chiamato Intel image classification.

Il dataset è già stato diviso in training set e test set. Il training set è suddiviso al suo interno in altre 6 cartelle per ogni categoria di immagine *buildings*, *forest*, *glacier*, *mountain*, *sea*, *street*, per l'allenamento della rete si è deciso di utilizzare 2000 immagini per cartella, di conseguenza 12000 immagini mentre 190 per cartella per la validazione. Qui sotto vengono riportate 6 immagini esemplificative di quello che troviamo in ognuna delle 6 cartelle.



Fig. 1: Esempi immagini appartenenti alle cartelle di training

Nel dataset in esame sono stati riscontrati alcuni problemi guardando le cartelle di training. Sono state infatti trovate immagini non appartenenti alla categoria attesa all'interno di quella cartella, un esempio viene qua sotto riportato:



Fig. 2: Immagine macchina presente nella cartella glacier

Abbiamo quindi optato per eliminare le immagini che non rispettavano l'etichetta posta a nome della cartella perchè queste immagini costituivano un ostacolo per l'apprendimento della nostra rete.

2.2 Metodi

L'analisi si è composta di diverse fasi, prima fase abbiamo preso le prime 2000 immagini per ogni sottocartella presente nella cartella di training e abbiamo creato il nostro training set, lo stesso abbiamo fatto per il validation set. Abbiamo creato 2 funzioni la prima che trasforma le immagini in tensori mentre la seconda che da un'etichetta a seconda della classe a cui appartiene la singola immagine. Una volta applicate le due funzioni abbiamo i nostri dati di test e di validation con le rispettive etichette.

Il passaggio successivo è stato quello di applicare il processo che viene detto di augmentation ovvero una tecnica utilizzata per migliorare la qualità dei dati utilizzati al fine di addestrare le reti neurali artificiali. L'augmentation implica l'aumento artificiale delle dimensioni del training set aggiungendo trasformazioni per lo più di carattere geometrico (anche se a volte possono riguardare l'immagine a livello di luminosità o nitidezza) ai campioni di dati esistenti.

Sia per il training set che per il validation che per il test set abbiamo riscalato le immagini in $[0,1]$ mentre per il training set abbiamo applicato delle trasformazioni che vengono riportate nella tabella qui sotto proposta:

Trasformazione	Valore scelto
width shift range	0.25
height shift range	0.25
shear range	0.25
zoom range	0.35
horizontal flip	true
fill mode	nearest

Table 1: Tabella parametri data augmentation

Vediamo ora cosa significhino ognuno di questi parametri:

1. *Width shift*: indica lo spostamento orizzontale.
2. *Height shift*: indica lo spostamento verticale.
3. *Shear range*: indica lo spostamento di un oggetto ad una distanza costante.
4. *Zoom range*: indica lo zoom sull'immagine.
5. *Horizontal flip*: capovolge le immagini orizzontalmente.
6. *Fill mode*: che sceglie come riempire i pixel vuoti, in questo caso si basa sul più vicino.

Una volta fatto ciò generiamo le nostre immagini di training sulla base delle trasformazioni, e riscaldiamo train, validation e test set.

Per quanto riguarda il modello si è deciso di utilizzare una rete neurale coinvoluzionale, questo perché estremamente adeguata all'immagine processing infatti la rete coinvoluzionale tramite le trasformazioni Kerne è in grado di ridurre drasticamente il numero di parametri in gioco.

Dopo vari tentativi di rete facendo variare il numero di layer di coinvoluzione, il numero di nodi per ciascun layer e alcuni iperparametri come il rate del layer dropout si è arrivati alla configurazione della rete di seguito presentata:

La rete neurale coinvoluzionale è composta da un *input layer* con funzione di attivazione *Relu* seguito da 3 layer di coinvoluzione il primo con 32 nodi mentre gli altri 2 ognuno con 128 nodi, ovviamente per tutti e 3 gli strati è stata predisposta una funzione di attivazione *Relu* come anche un Kernel di dimensione 3x3.

Dopo ogni layer di coinvoluzione è stato posto uno strato di pooling che ha il compito di ridurre gradualmente la dimensione della matrice ma di aumentare il livello di astrazione.

Successivamente è stato inserito un *layer dropout* che è necessario per evitare che il modello finisca nell'overfitting quello che il *dropout* fa è infatti che alcuni dei neuroni negli strati nascosti o visibili vengono rilasciati o omessi casualmente. Nel *layer dropout* va impostato un parametro ovvero il *rate* che indica la frazione delle unità da dropare che abbiamo deciso di impostare pari a 0.43.

Infine è stato aggiunto un livello *fully connected* che poi esegue di fatto la classificazione, siccome il problema è di classificazione multiclasse è stata scelta una funzione di attivazione *softmax*.

Si è deciso di massimizzare l'*accuracy* minimizzando come funzione di perdita la *categorical crossentropy* che è una funzione di perdita utilizzata nelle task di

classificazione multiclasse, la funzione è costituita in maniera tale per cui venga calcolata una distanza tra due distribuzioni di probabilità.

Per ottimizzare i pesi si è utilizzato un ottimizzatore *adam* con learning rate posto pari a $1e-3$.

L'allenamento della rete è stato poi effettuato su 30 epoche e per ogni epoca sono stati impostati 100 step.

3 Risultati

Per il modello preso in esame è stata monitorata la *loss* e la *accuracy* sia sul training set che sul validation set, ricordiamo come il nostro obiettivo sia quello di ottenere la migliore classificazione possibile sul test set, privilegiamo dunque avere un valore elevato sul validation set rispetto ad uno sul training set siccome il validation set è una simulazione di quello che può accadere a livello di classificazione nel test set.

Nel grafico di seguito presentato viene mostrato l'andamento *loss* e la *accuracy* sia sul training set che sul validation set del modello in esame:

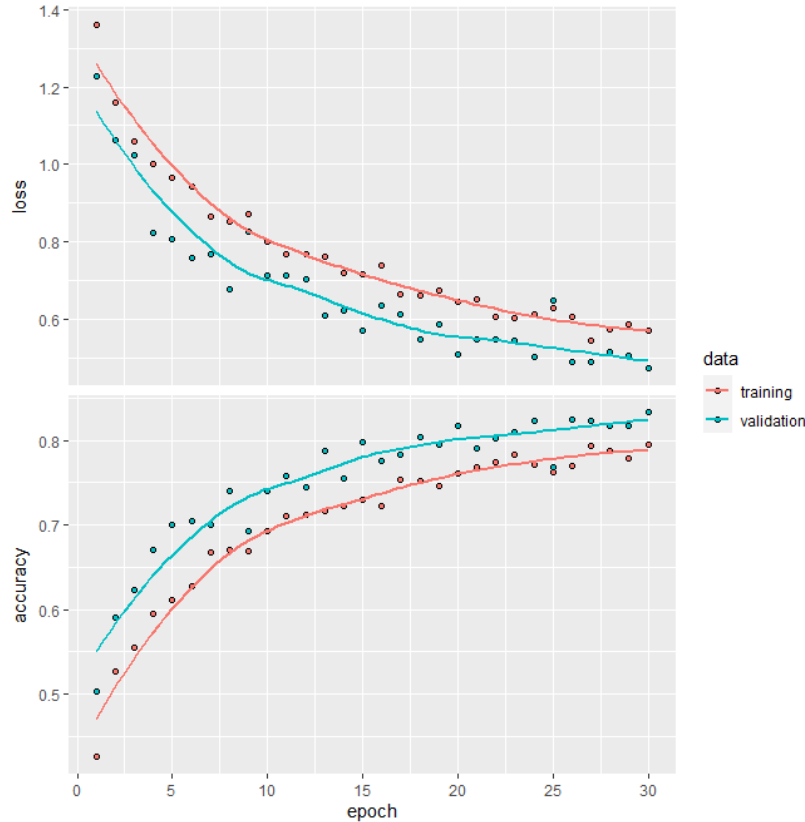


Fig. 3: Loss e Accuracy sul traing e sul validation set

Come possiamo notare dal grafico la classificazione funziona meglio sul validation set rispetto al training set e possiamo affermare anche chiaramente che il modello non va incontro al fenomeno dell'overfitting.

Diamo ora uno sguardo alla matrice di confusione presentata di seguito per verificare dove gli errori avvengono nello specifico.

Risulta abbastanza evidente che la maggior parte degli errori si concentrino nella

	buildings	forest	glacier	mountain	sea	street
buildings	378	4	8	14	17	66
forest	6	461	5	3	9	5
glacier	3	0	363	59	9	0
mountain	2	1	52	384	11	2
sea	7	1	72	52	451	2
street	41	7	35	12	12	426

Table 2: Matrice di confusione test set

classificazione delle immagini della classe *glacier*, non esiste una classe con cui la confonda in maniera più evidente rispetto alle altre ma le classi che creano problemi nella classificazione sono *mountain* e *sea*.

Di seguito viene ripotata un'immagine appartenente al test set della classe *glacier*:



Fig. 4: Immagine test set classe glacier

Il ghiacciaio come mostrato in figura 4 scende da una montagna e spesso da vita a specchi d'acqua simili al mare pertanto la rete fatica a distinguere tra le 3 classi poichè spesso sono presenti nella stessa immagine contemporaneamente oltre al fatto che possono avere colori molto simili (le montagne ricoperte di neve hanno lo stesso colore del ghiacciaio e il mare è azzurro e spesso può essere confuso con un ghiacciaio).

Vengono commessi errori anche nella classificazione della classe *mountain* presochè per lo stesso motivo.

La rete tende inoltre in alcuni casi a confondere *buildings* con *street* nonostante il modello le distingua molto bene.

Risulta invece ottima la classificazione della classe *forest*.

4 Discussioni

L'obiettivo dell'analisi era quello di prevedere raggiungendo il più alto valore possibile di accuracy i dati appartenenti al test set di Intel Image classification, un dataset contenente immagini appartenenti a 6 classi differenti.

Nello specifico il dataset era già suddiviso in training e test set, il primo sarebbe servito per l'allenamento della rete, mentre la classificazione sul test set era il reale obiettivo dello studio.

Si è proposto un modello di rete neurale coinvoluzionale con 3 strati di coinvoluzione che sul validation set, ottenuto togliendo una frazione di dati al training set ha ottenuto un valore di accuracy dell'0.82 confermato poi sul test set. Abbiamo poi visto grazie ad una matrice di confusione dove si concentrassero gli errori e abbiamo provato a spiegare cosa stesse alla base di questi errori che la rete continuava a commettere, si trattava infatti di dati molto dubbi su cui la rete faticava a prendere una decisione univoca.

Una possibile soluzione che tuttavia non si è riusciti ad affrontare nell'analisi presentata è quello di non lavorare con esclusivamente 3 canali ma aggiungere più canali in maniera tale per cui si possano apprendere meglio le peculiarità dell'immagine ed essere ancor più precisi in materia di classificazione sul test set.