Advanced Machine Learning - Assignment 4

Luca Gandolfi, matricola 807485

Novembre 2019

1 Introduzione

L'assignment consiste nel testare il Transfer Learning attraverso la tecnica del Feature Extraction. In questo report verranno analizzate le scelte effettuate durante il design della rete, a partire dalla scelta del modello *pre-trained* e del nuovo task di apprendimento, per poi analizzare diversi livelli di taglio e confrontare i risultati della classificazione con Support Vector Machine.

2 VGG-16: Modello pre-trained

Come modello pre-trained, tra quelli presenti nella libreria Keras è stato scelto il modello VGG-16. In particolare, questo modello è uno dei più semplici (e di conseguenza veloci) a risolvere con buone performance la classificazione del dataset ImageNet, contenente immagini appartenenti a ben 1.000 classi diverse.

Data la numerosità delle classi del dataset originale, ci aspettiamo che siano contenute anche classi relative al nuovo task di classificazione, il quale verrà discusso nel prossimo paragrafo.

Il modello VGG-16 è stato quindi utilizzato con i pesi ImageNet forniti da Keras e, in tre esperimenti diversi, è stato tagliato dopo blocchi convoluzionali diversi per ottenere nuove feature e confrontare le performance sul nuovo task di classificazione.

3 Nuovo task di classificazione

3.1 CIFAR-10 e riduzione della dimensionalità

Il dataset utilizzato per il nuovo task di apprendimento è un sottoinsieme del CIFAR-10, un dataset molto famoso contenente 60.000 immagini 32x32 a colori (quindi 32x32x3) suddivise in 10 classi e con 10.000 di queste dedicate al test set. In particolare, le classi sono bilanciate in quanto ogni classe nel train presenta 5.000 esempi e ogni classe del test set presenta 1.000 esempi. Le classi contenute nel dataset sono: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck.

Di queste classi, soltanto 4 sono state selezionate per essere utilizzate nel nuovo task: Airplane, Bird, Dog, Horse. L'idea originale era quella di considerare un numero di esempi ristretto, come ad esempio soltanto 2.000 immagini per classe, tuttavia i risultati migliori sono stati ottenuti considerando tutte le 5.000 immagini del train set per le rispettive classi. Per il test set invece, sono state utilizzate soltanto 500 immagini per classe.

La scelta delle classi ha avuto anche lo scopo di mettere in difficoltà il modello: aereo e uccello, in genere, sono immagini visivamente simili, così come anche il cavallo potrebbe vagamente ricordare un cane più grande. Probabilmente, scegliere classi totalmente diverse tra loro come ad esempio Airplane, Frog, Horse, Truck avrebbe portato a risultati migliori in quanto le immagini presentano meno caratteristiche in comune.

3.2 Analisi preliminari del nuovo dataset

Come è stato anticipato, sono stati costruiti train e test set bilanciati. Per facilitare la lettura di seguito è riportata una tabella riassuntiva delle classi utilizzate nel nuovo task e del relativo numero di record.

Classe	# Record Train	# Record Test
Airplane	5.000	500
Bird	5.000	500
Dog	5.000	500
Horse	5.000	500

Tabella 1: Distribuzione dei record: dataset perfettamente bilanciato.

Data la numerosità del problema originale, ci si aspetta che le classi scelte appartengano anche al dataset ImageNet e di conseguenza ci si aspetta che il modello sia in grado di ottenere delle buone feature dalle immagini presenti nel nuovo dataset. Le performance potrebbero però essere limitate da, principalmente, due fattori: l'utilizzo di un modello *basic* come SVM e il ridotto numero di esempi. Anche la differenza sostanziale presente nel numero di classi con il problema originale potrebbe influenzare la scelta della posizione di estrazione delle nuove feature.

In Figura 1, è possibile osservare le prime 5 immagini del train e del test set. Le immagini sono 32x32 quindi molto piccole e tendono a sgranarsi se zoommate per essere viste dall'occhio umano. Per l'algoritmo tuttavia questo non sarà un problema. E' possibile inoltre notare come le foto siano appositamente molto diverse e talvolta presentino anche elementi di disturbo, come ad esempio il fantino sopra il cavallo nella terza immagine della prima riga, o la scritta in rosa nella prima immagine della seconda riga.

Figura 1: Prime 5 immagini nel train set (sopra) e prime 5 immagini nel test set (sotto).

4 Feature Extraction: scelte di design

Come anticipato nel precedente paragrafo, la presenza di solo 4 classi rispetto alle 1.000 del problema originale potrebbe tendenzialmente allontare una similitudine tra i due task, prediligendo un taglio più vicino all'input. D'altro canto, la presenza di classi che, probabilmente, sono presenti anche in ImageNet potrebbe portare a prediligere un taglio vicino all'output. E' stato scelto, quindi, di confrontare i risultati ottenuti con tre punti di taglio diversi: al termine dei blocchi di convoluzione, sostituendo la parte fully-connected con un modello SupportVector Machine, prima dell'inizio dell'ultimo blocco di convoluzione e infine dopo il terzo blocco di convoluzione.

VGG16 presenta in totale 5 blocchi di convoluzione, seguiti da un blocco fully-connected utilizzato per la classificazione. Dato che i tagli sono stati effettuati tutti prima della parte fully-connected, l'output dei blocchi di convoluzione è una matrice. I modelli sono quindi stati aggiornati dopo il taglio con il metodo *Flatten*, che ha permesso di ottenere una rappresentazione vettoriale delle matrici. Non è stato necessario allenare i modelli poiché sono stati generati con già i pesi di ImageNet impostati. In Figura 2 è possibile osservare graficamente la posizione dei tagli rispetto ai blocchi convoluzionali.

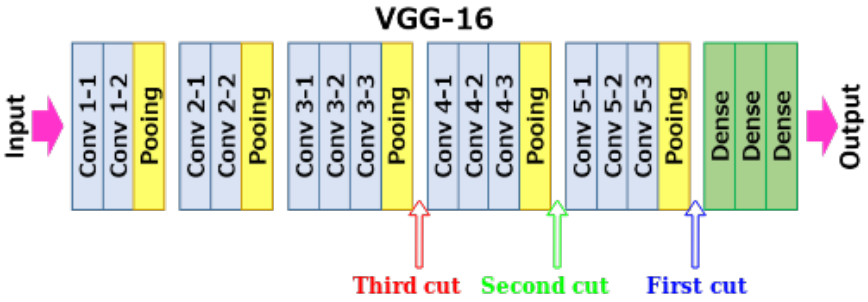

Figura 2: Posizione dei tagli per Feature Extraction su VGG-16.

Man mano che ci si avvicina ai layer densi, il numero di feature estratte sarà ridotto. Nella seguente tabella è possibile osservare il numero di feature estratte dalle immagini in base al punto di taglio. Ovviamente, il numero di feature è calcolato dopo aver applicato la Flatten alle matrici.

Output	# Feature
Conv5_pool	512
Conv4_pool	2.048
Conv3_pool	4.096

Tabella 2: Feature estratte in base al punto di taglio.

5 Classificatore standard

Come classificatore standard sono stati presi in esame due modelli differenti: k-NN e SVM. Dati gli scarsi risultati ottenuti con k-NN, è stato deciso di utilizzare SVM.

SupportVector Machine consiste in un insieme di tecniche supervisionate per classificazione, regressione e individuazione di outliers. In questo caso, il task consiste in classificazione e la libreria *sklearn* fornisce due metodi principali per la classificazione multi-classe: SVC e LinearSVC. Entrambi i metodi sono stati testati e, nonostante i tempi di elaborazione decisamente più lunghi, SVC è stato preferito in quanto fornisce performance leggermente più alte.

SVC è stato definito con i parametri standard consigliati nella documentazione, con particolare focus sulla funzione di decisione di tipo *one-versus-one*, generalmente utilizzato per classificazioni multi-classe rispetto al tipo *one-versus-rest*. Questa impostazione permette la creazione di più classificatori, ognuno dei quali va a classificare soltanto due classi per volta.

SVC è stato quindi utilizzato per classificare i dati di train dopo essere stati processati dai blocchi convoluzionali. L'input non è una matrice 32x32(x3) ma semplicemente un vettore rappresentante le feature estratte tramite Transfer Learning, ottenuto grazie all'aggiunta della Flatten in coda ai blocchi di convoluzione.

Riassumendo, il funzionamento della tool-chain costruita è il seguente:

1. Estrazione di VGG-16 fino al punto di taglio in esame;
2. Aggiunta in coda al modello di una Flatten;
3. Predict del train utilizzando i pesi di ImageNet, ottenendo nuove feature rispetto al problema originale;
4. Predict anche sul test set, per ottenere lo stesso formato del train;
5. Apprendimento con SVM del nuovo train set ottenuto dopo la predict del modello pre-trained;
6. Valutazione delle performance utilizzando il nuovo test set.

6 Analisi dei risultati

In questo paragrafo sono trattati separatamente i tre esperimenti svolti. Per ogni esperimento, chiamato taglio poiché caratterizzato da un punto di taglio diverso sul modello VGG-16, è mostrata prima la distribuzione degli esempi nello spazio tramite t-SNE, e poi le performance ottenute applicando il classificatore standard. Le performance valutate sono state molteplici: matrice di confusione, accuracy, precision e recall.

Una piccola nota: osservando i plot di t-SNE è facile intuire il motivo per cui k-NN abbia riscontrato performance non soddisfacenti. k-NN, infatti, assegna un record ad una classe basandosi sulla sua posizione nello spazio, analizzando un certo numero di vicini. In tutti e tre gli esperimenti, i record si sono rivelati molto vicini tra loro nello spazio, portando quindi ad un elevato numero di errori.

6.1 Primo taglio

Il primo taglio a VGG-16 è stato effettuato dopo il blocco convoluzionale 5, visibile in Figura 2. In questo caso i record erano composti da 512 feature. Come emerge dalla proiezione bi-dimensionale con t-SNE, i record sono molto sovrapposti tra loro, sintomo di una compressione troppo elevata.

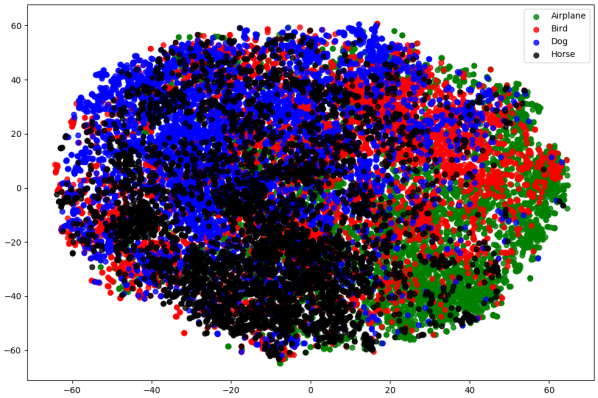

Figura 3: Proiezione delle feature ottenute dopo l'ultimo blocco di convoluzione di VGG-16.

Come facilmente ipotizzabile, data la compressione (del numero di feature) troppo elevata, i risultati applicando SVM sono stati pessimi. In Figura 4 è mostrata la matrice di confusione, che indica le percentuali normalizzate di record correttamente classificati (in tonalità di blu a seconda del valore) e la distribuzione di errori. Generalmente, si osserva la diagonale in blu, tuttavia può essere interessante osservare anche come sono distribuiti gli errori. In questo caso è possibile notare come Airplane venga in linea di massima classificato correttamente e, il 7% di record di questa classe venga classificato come Bird. Non si può dire lo stesso per Bird, che viene sbagliato spesso e, in particolare, viene scambiato per il 13% dei record per Dog.

In Tabella 3 è invece mostrato il classification report, raffigurante per ogni classe i valori di Precision, Recall e f1-Measure.

Classe	Precision	Recall	f1-Score
Airplane	0.81	0.85	0.83
Bird	0.73	0.69	0.71
Dog	0.73	0.76	0.74
Horse	0.77	0.74	0.76

Tabella 3: Metriche Precision, Recall, f1-Measure per classe.

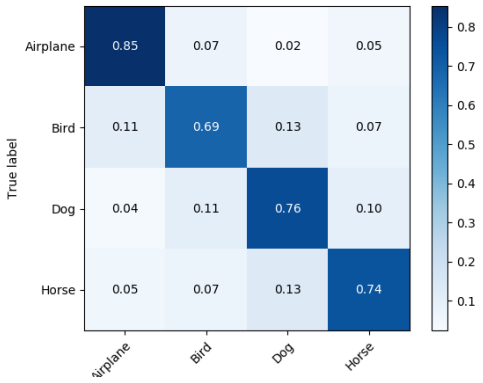

Figura 4: Confusion matrix relativa alle performance ottenute con le feature estratte dopo l'ultimo blocco di convoluzione di VGG16.

Concludendo, tagliando la CNN VGG-16 dopo il quinto blocco convoluzionale è stata ottenuta una **Accuracy** pari a **0.7605**, un valore buono ma non sufficiente visto le capacità della rete. Si può fare di meglio, provando a considerare più feature.

6.2 Secondo taglio

Il secondo taglio a VGG-16 è stato effettuato dopo il quarto blocco di convoluzione, visibile in Figura 2. In questo caso le feature rappresentanti le immagini erano 2048. Come è possibile osservare dalla proiezione in due dimensioni di t-SNE, i record sono meno sparsi rispetto alla situazione precedente. 3 classi su 4, Horse, Dog e Airplane sono ben compatte in zone dense, presentando comunque delle eccezioni. La classe Bird risulta invece ancora molto problematica. Un'ulteriore osservazione sul grafico può essere fatta osservando la distanza tra le classi Horse e Airplane, come si può immaginare, le immagini di questi due soggetti sono molto diverse tra loro e queste differenze hanno permesso di far emergere feature tra loro distanti.

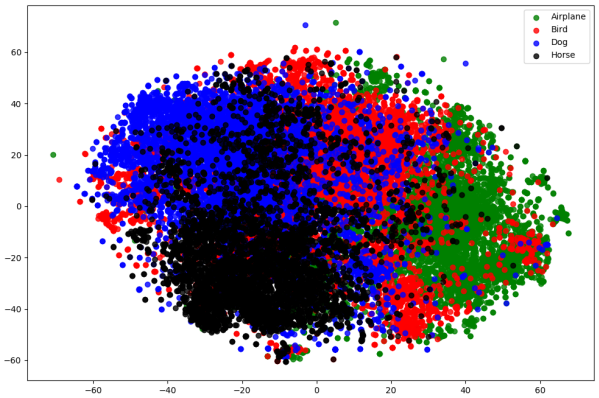

Figura 5: Proiezione delle feature ottenute dopo il quarto blocco di convoluzione di VGG-16.

Come per il paragrafo 6.1, vengono ora mostrati la matrice di confusione e il classification report. I risultati sono stati nettamente migliori rispetto all'esperimento precedente. Una nota particolare riguarda le performance sulla classe Bird, che nonostante la sparsità dei vettori ha comunque ottenuto l'85% di Recall.

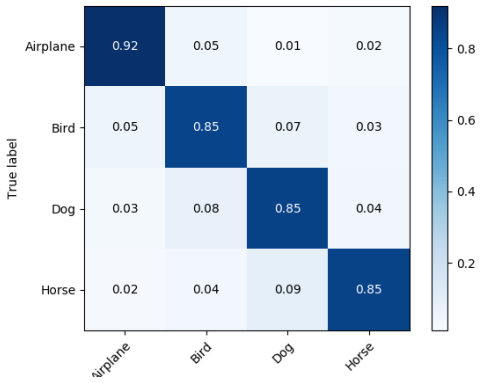

Figura 6: Confusion matrix relativa alle performance ottenute con le feature estratte dopo il quarto blocco di convoluzione di VGG16.

I miglioramenti sono osservabili anche confrontando il classification report. Le classi Bird e Dog restano quelle più difficili da classificare.

Classe	Precision	Recall	f1-Score
Airplane	0.90	0.92	0.91
Bird	0.84	0.85	0.84
Dog	0.83	0.85	0.84
Horse	0.90	0.85	0.87

Tabella 4: Metriche Precision, Recall, f1-Measure per classe.

Concludendo, tagliando la CNN VGG-16 dopo il quarto blocco convoluzionale è stata ottenuta una **Accuracy** pari a **0.8675**, un valore molto più alto rispetto all'esperimento precedente. Considerando come l'utilizzo di più feature abbia portato a migliori risultati, il prossimo taglio andrà ad utilizzare il doppio delle feature utilizzate in questo esperimento.

6.3 Terzo taglio

Il terzo taglio a VGG-16 è stato effettuato dopo il terzo blocco di convoluzione, visibile in Figura 2 con il colore rosso.

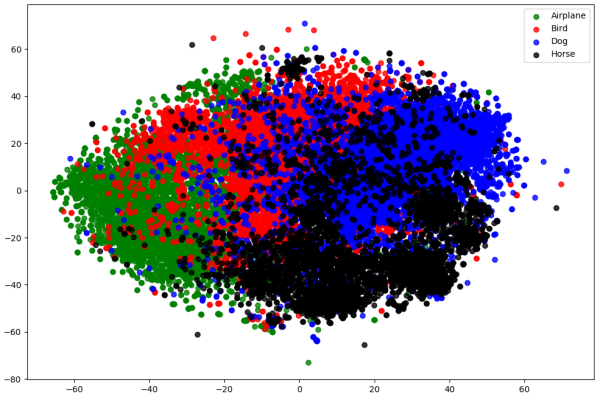

Figura 7: Proiezione delle feature ottenute dopo il terzo blocco di convoluzione di VGG-16.

In questo esperimento i tempi di elaborazione si sono rivelati molto lunghi, dato prevedibile considerando le 4096 feature. Osservando le proiezioni in due dimensioni di t-SNE, non emergono differenze significative con il secondo teglio, se non per la classe Bird che ora è meno sparsa. Di contro, la classe Horse ha perso la sua forte compattezza e si è un po' più appiattita. Come ipotizzato a inizio report, le classi Airplane e Bird continuano ad essere molto vicine e a tratti sovrapposte, data la somiglianza in molti casi delle due immagini (tralasciando i colori, entrambi con le ali aperte).

Basandosi esclusivamente sul grafico di t-SNE, non è possibile aspettarsi risultati migliori, se non per la classe Bird. Invece, il modello SVM ha riscontrato performance migliorate su tutte le classi tranne proprio Bird, che ha subito un leggero calo sulla Recall e migliorato leggermente la Precision.

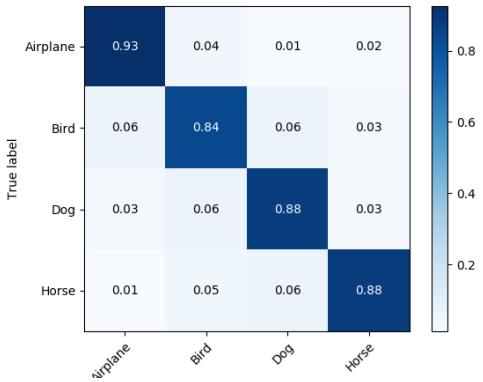

Figura 8: Confusion matrix relativa alle performance ottenute con le feature estratte dopo il terzo blocco di convoluzione di VGG16.

Anche questa volta la classe con performance migliori risulta essere Airplane. L'aumento delle feature nel corso degli esperimenti ha beneficiato molto la classe Horse, che passa da una Recall di 0.74 a 0.88. La classe Bird è quella migliorata più di tutti, toccando il suo apice però nel secondo esperimento.

Classe	Precision	Recall	f1-Score
Airplane	0.90	0.93	0.91
Bird	0.85	0.84	0.84
Dog	0.86	0.88	0.87
Horse	0.92	0.88	0.90

Tabella 5: Metriche Precision, Recall, f1-Measure per classe.

Concludendo, tagliando la CNN VGG-16 dopo il terzo blocco convoluzionale è stata ottenuta una **Accuracy** pari a **0.881**, un valore che migliora l'accuracy ottenuta nel secondo esperimento. Bisogna tener presente però i tempi di elaborazione richiesti, che diventano esagerati per la proiezione di t-SNE toccando circa 50 minuti di esecuzione.

7 Conclusioni

L'utilizzo di un classificatore standard come SupportVector Machine non ha permesso di ottenere risultati superiori al 90% di Accuracy, tuttavia nel terzo esperimento, estraendo le feature dopo 3 blocchi convoluzionali di VGG-16, è stato possibile raggiungere l'88.1%, valore più che soddisfacente. Il Transfer Learning sicuramente permette di velocizzare l'apprendimento per situazioni

simili a quella trattata, bisogna precisare però che per ottenere risultati davvero ottimi sarebbe necessario l'utilizzo di modelli più complessi a cui dare in pasto le feature estratte, come ad esempio reti deep fully-connected.

Tornando ad analizzare i risultati ottenuti, contro l'ipotesi iniziale che avrebbe suggerito di preferire tagli più vicini all'output, i risultati migliori sono stati ottenuti tagliando VGG-16 circa a metà, dopo il terzo blocco convoluzionale. La supposizione ovvia per giustificare questi risultati è il numero di feature troppo compresso soprattutto nell'ultimo blocco (Primo esperimento), ovvero 512. Il modello per funzionare bene ha bisogno di più feature. Avvicinarsi ulteriormente all'input, ad esempio tagliando dopo il secondo blocco convoluzionale, avrebbe in realtà aumentato di molto i tempi di elaborazione producendo un numero troppo elevato di feature, che sappiamo essere un problema per i classificatori. Per questo motivo questo esperimento è stato scartato.

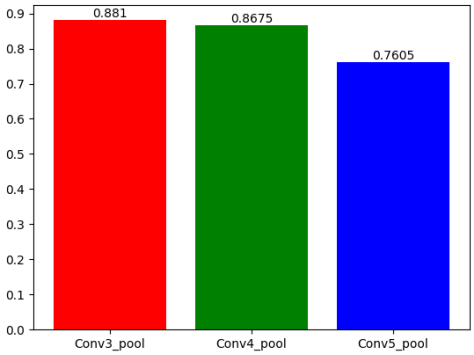

Figura 9: Confronto tra le Accuracy ottenute durante i tre esperimenti.

In Figura 9, è possibile osservare graficamente l'andamento della Accuracy a seconda dei tagli. I colori rimandano quelli utilizzati in Figura 2 per descrivere la posizione dei tagli. L'asse delle ascisse indica semplicemente il layer dopo il quale sono state estratte le feature.