



Assignment IV

1920-2-F1801Q151, Advance Machine Learning, 2019-2020

Università degli studi di Milano Bicocca

Dipartimento di Informatica, Sistemistica e Comunicazione

Villa Giacomo 807462

25 Novembre 2019

Indice

1	Dataset e Operazioni preliminari	3
2	Transfer Learning	4
2.1	Features Extraction	4
2.2	Support Vector Machine	5
2.3	Valutazione delle performance	5
3	Conclusioni	8

1 Dataset e Operazioni preliminari

Per l'assignment è stato richiesto di implementare una pipeline di **transfer learning** con approccio **feature extraction**. Partendo da una rete pre-trainata si è andato ad effettuare diversi tagli, costruendo di fatto diversi modelli, al fine di **estrarre una nuova rappresentazione del dato posto in input**; la scelta di quest'ultimo, parte integrante dell'assignment stesso, è ricaduta su un **sottoinsieme del dataset CIFAR-10**. In particolare, ho selezionato **quattro classi** (Horse, Ship, Frog e Airplane) ottenendo di fatto il dataset successivamente utilizzato per l'assignment stesso; un **esempio di elementi da classificare** è il seguente:

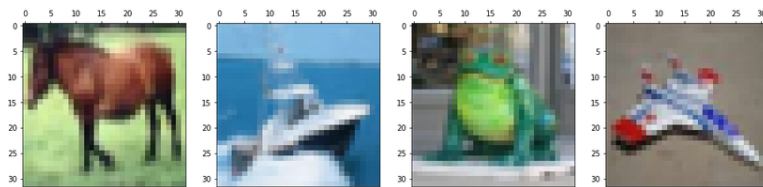


Figura 1: Esempio record da classificare

Dunque l'**immagine** è di fatto una matrice $32 \times 32 \times 3$ (immagine a colori) **ogni cella** detiene un valore numerico che di fatto rappresenta l'**intensità luminosa del pixel** in questione (range $[0, 255]$). Al fine di permettere un corretto funzionamento è stato effettuato un **preprocessing** dove si è provveduto a **riscalare i valori** delle celle in un range $[0, 1]$. Il **training set** è composto da **20000 immagini** dotate di etichette e il **test set** è composto da **4000 immagini** con etichetta al fine di permettere il calcolo delle performance. È stato eseguito uno **studio sulla distribuzione degli elementi**, lo stesso ha prodotto il seguente risultato:

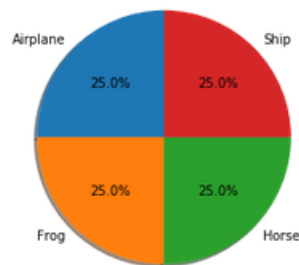


Figura 2: Distribuzione classi nel training set

Data la **situazione completamente bilanciata** è stato ritenuto lecito effettuare

un confronto, dati i diversi tagli, completamente basato sull'accuracy. Il test set è composto da 1000 immagini per categoria.

2 Transfer Learning

2.1 Features Extraction

La metodologia di Transfer learning scelta prende il nome di *Features Extraction*, sono partito dalla rete VGG16 pre-addestrata sul dataset ImageNet; questo è dotato di oltre 20.000 categorie con una dimensione pari a 14 milioni di immagini. I tagli effettuati sono i seguenti:

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32, 32, 3)	0
block1_conv1 (Conv2D)	(None, 32, 32, 64)	1792
block1_conv2 (Conv2D)	(None, 32, 32, 64)	36928
block1_pool (MaxPooling2D)	(None, 16, 16, 64)	0
block2_conv1 (Conv2D)	(None, 16, 16, 128)	73856
block2_conv2 (Conv2D)	(None, 16, 16, 128)	147584
block2_pool (MaxPooling2D)	(None, 8, 8, 128)	0
block3_conv1 (Conv2D)	(None, 8, 8, 256)	295168
block3_conv2 (Conv2D)	(None, 8, 8, 256)	590080
block3_conv3 (Conv2D)	(None, 8, 8, 256)	590080
block3_pool (MaxPooling2D)	(None, 4, 4, 256)	0
block4_conv1 (Conv2D)	(None, 4, 4, 512)	1180160
block4_conv2 (Conv2D)	(None, 4, 4, 512)	2359808
block4_conv3 (Conv2D)	(None, 4, 4, 512)	2359808
block4_pool (MaxPooling2D)	(None, 2, 2, 512)	0
block5_conv1 (Conv2D)	(None, 2, 2, 512)	2359808
block5_conv2 (Conv2D)	(None, 2, 2, 512)	2359808
block5_conv3 (Conv2D)	(None, 2, 2, 512)	2359808
block5_pool (MaxPooling2D)	(None, 1, 1, 512)	0

=====

Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0

Figura 3: Tagli effettuati

Ad ogni taglio coincide la creazione di un nuovo modello, in funzione del taglio si otterrà una nuova rappresentazione del dato in input, appiattendolo la

stessa (funzione `flatten()`) si ottiene di fatto un vettore per ogni input che altro non è che una nuova rappresentazione dell'immagine. In particolare, **dati i cinque tagli** proposti, si otterranno dei vettori con le **seguenti dimensioni**:

- 16384 elementi
- 8192 elementi
- 4096 elementi
- 2048 elementi
- 512 elementi

In funzione del taglio **l'immagine sarà descritta da un certo numero di features** mostrato nell'elenco; ho deciso di **tagliare in prossimità dei layers di MaxPooling** al fine di avere per ogni taglio rappresentazioni di dimensioni diverse dell'input. Data una nuova rappresentazione è stato poi possibile allenare un **modello di Machine Learning “tradizionale”** al fine di adempiere al problema di classificazione posto; in particolare si è scelto di utilizzare una **SVM offerta dalla libreria *Scikit-learn***; in particolare è stata utilizzata SVC che permette classificazione multiclasse.

2.2 Support Vector Machine

La SVM allenata è dotata di un **kernel lineare**, questa scelta è stata presa in quanto i tempi computazionali richiesti dal modello stesso risultano essere, in questo modo, inferiori rispetto a all'utilizzo di altri kernel. **Le performance registrate dimostrano che gli elementi appartenenti alla differenti classi risultano essere, nella maggior parte dei casi, linearmente separabili**; inoltre, data la dimensione delle features, ho preferito utilizzare un separatore lineare.

I **valori degli altri iperparametri disponibili non sono stati alterati** in quanto le performance non ne vedevano un significativo aumento. Per tanto è possibile affermare che **la configurazione di default si è dimostrata ottimale** per il problema di classificazione.

2.3 Valutazione delle performance

Al fine di **valutare le performance** si è provveduto ad **allenare**, in funzione del taglio, una SVM **con i 20000 dati di training**; gli stessi, in funzione del taglio,

erano dotati di una diversa rappresentazione.

In seguito all'allenamento dei modelli si è poi provveduto a valutare le performance quali *Accuracy*, *Precision*, *Recall* e *F1-Score* per ogni possibile configurazione. I risultati sono dunque i seguenti:

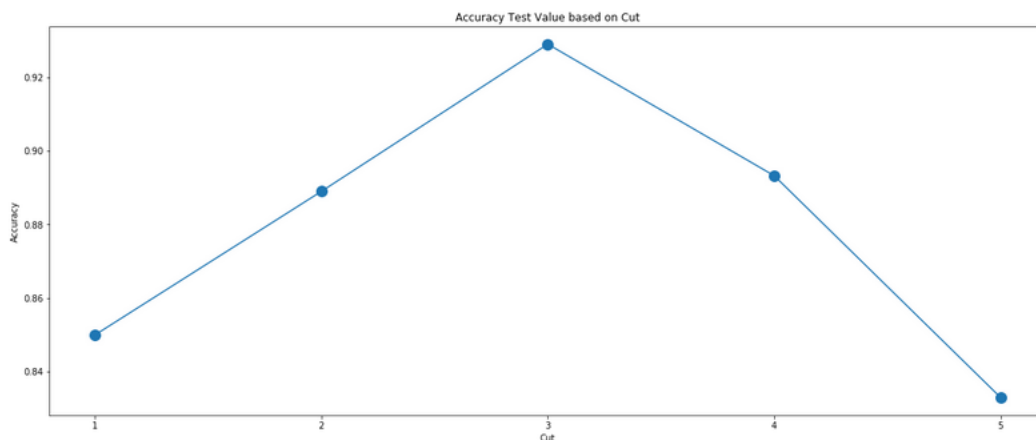


Figura 4: Accuracy in funzione del taglio

Precision					
	Cut 1	Cut 2	Cut 3	Cut 4	Cut 5
Airplane	0.77	0.81	0.89	0.83	0.78
Frog	0.88	0.94	0.96	0.93	0.88
Horse	0.91	0.92	0.95	0.93	0.87
Ship	0.84	0.89	0.92	0.89	0.80

Recall					
	Cut 1	Cut 2	Cut 3	Cut 4	Cut 5
Airplane	0.81	0.86	0.91	0.86	0.77
Frog	0.92	0.93	0.95	0.93	0.89
Horse	0.87	0.92	0.95	0.92	0.86
Ship	0.81	0.85	0.92	0.87	0.81

F1-Score					
	Cut 1	Cut 2	Cut 3	Cut 4	Cut 5
Airplane	0.79	0.84	0.90	0.85	0.78
Frog	0.90	0.93	0.95	0.93	0.89
Horse	0.89	0.92	0.95	0.92	0.86
Ship	0.82	0.87	0.92	0.88	0.81

Risulta possibile osservare come le performance tendano a salire in concomitanza al terzo taglio per poi calare andando a registrare la peggior performance presso l'ultimo taglio; registrando il **miglior risultato presso il terzo taglio** si può evincere che la classificazione di Airplane, Frog, Horse e Ship non si discosta in maniera eccessiva dal task originale della rete VGG16.

Come **ulteriore prova delle performance** sopra riportate fornisco le **matrici di confusione** calcolate, ovviamente, in funzione dei tagli applicati:

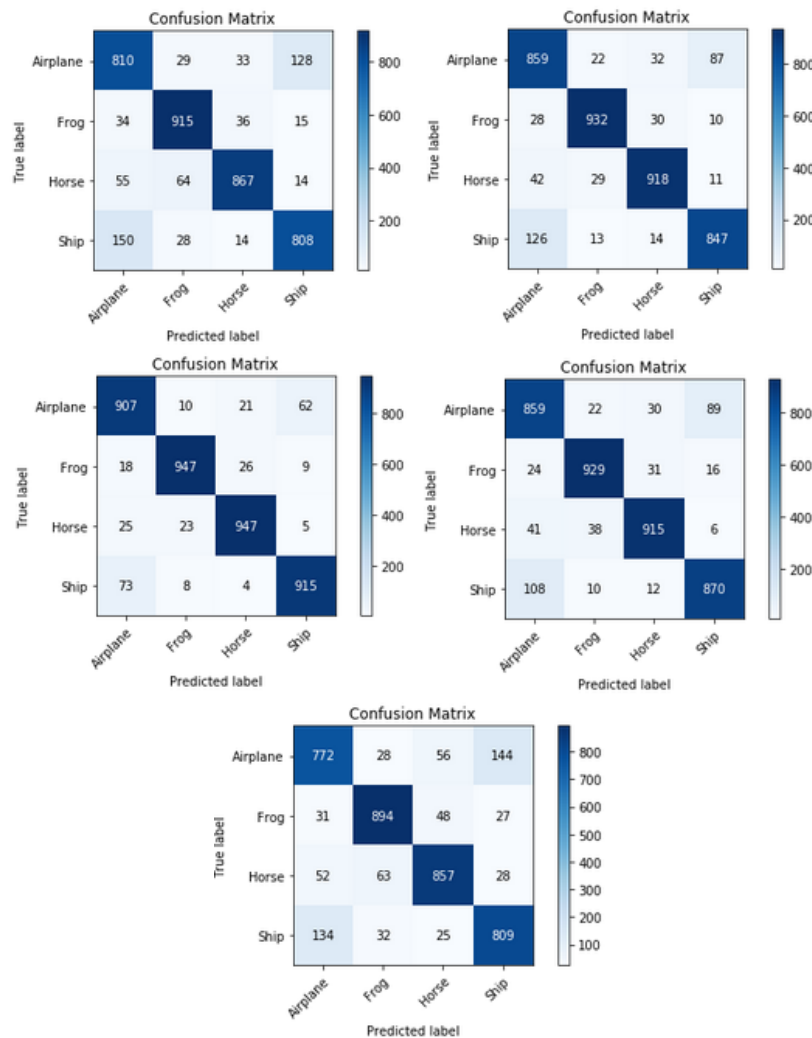


Figura 5: Matrice di confusione in funzione del 1°, 2°, 3°, 4° e 5 taglio

Possiamo notare come, in generale, la **problematica principale possa essere la confusione tra la classe *Ship* e la classe *Airplane***; gli elementi delle suddette classi presentano features comuni che causano l'errore. Una simile situazione, seppur molto più attenuata, può essere registrata per le classi *Frog* e *Horse*. In generale le **features estratte dal terzo taglio forniscono**, grazie alla SVM indotta, **i risultati migliori** per il task di classificazione.

Sulla configurazione migliore, quella ottenuta mediante il sul terzo taglio, **si registrano in totale 284 errori su un totale di 4000 elementi di test** (circa il 7%); il modello indotto sul terzo taglio è pertanto da ritenersi il migliore e quello indicato per effettuare il task richiesto.

3 Conclusioni

In generale i **risultati risultano essere soddisfacenti dato il task di classificazione**, l'unica problematica risulta essere la confusione tra le due classi rappresentanti i mezzi di trasporto.

Sarebbe **interessante indurre un altro modello**, differente dalla SVM, per **confrontare i risultati sui differenti tagli** al fine di osservare possibili miglioramenti degli stessi; inoltre un **confronto delle performance con la fine tuning** risulterebbe altrettanto interessante.