

Week 4 — Assignment Submission

Gianluca Scarpellini - 807541 - g.scarpellini1[at]disco.unimib.it
(In team con Federico Belotti - 808708)

24 novembre 2019

Indice

1 Task	1
2 Dataset e preprocessing	2
3 Modelli di feature extraction	3
3.1 VGG16	3
3.2 ResNet50	3
4 Esperimenti	5
4.1 Risultati ResNet50 - SVM	6
4.2 Risultati VGG16 - SVM	6
5 Conclusione	7
Bibliografia	9

1 Task

Il task oggetto di questo assignment riguarda l'extraction di features tramite modelli di Deep Learning pre-addestrati su Imagenet [Den+09]. Abbiamo scelto di impiegare due diversi modelli: VGG16 e ResNet50. I due modelli sono stati tagliati a diversi livelli e impiegati come feature-extractor. Successivamente, abbiamo impiegato un modello **SVM Lineare** quale classificatore su 5 classi. Il dataset addotato per il training e la validazione del è il famoso CIFAR-10 [KNH], in particolare per le classi: dog, frog, horse, ship, truck. Ho analizzato il dataset e il preprocessing nella sezione 2. Nella sezione 3 analizzo i modelli di CNN impiegati per l'estrazione di feature, in particolare nei punti in cui abbiamo deciso di effettuare i tagli. In 4 riporto i risultati ottenuti dal classificatore.

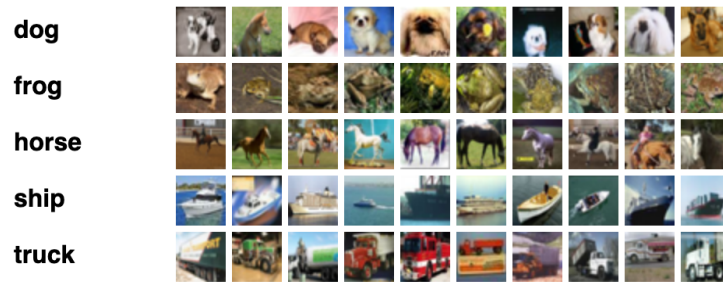


Figura 1: CIFAR5

2 Dataset e preprocessing

Il dataset impiegato per l'addestramento è il Cifar, ridotto a sole 5 classi. Il dataset consta di 25,000 immagini a colore, di cui 20,000 per il training e 5,000 per il test. Abbiamo deciso di estrarre dal training set un subset di immagini (che chiameremo val set) con cui valutare la bontà della scelta degli ipermateri. In fase di **feature extraction**, quando è stato impiegato il modello VGG abbiamo applicato il relativo preprocessing. Abbiamo inoltre optato per una normalizzazione di ciascun set di feature estratte, sottraendo per media e dividendo per deviazione standard dello stesso al fine di centrare il dataset in zero con varianza 1. In 2 riporto l'istogramma della distribuzione del dataset per classe. Questo risulta essere ben bilanciato.

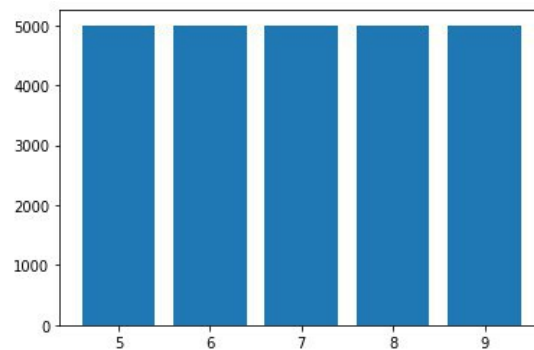


Figura 2: Distribuzione delle cifre del dataset CIFAR

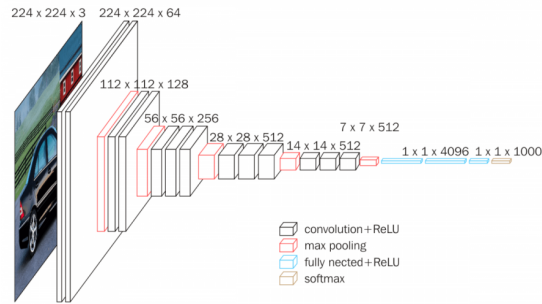


Figura 3: VGG16

3 Modelli di feature extraction

3.1 VGG16

Il primo modello di CNN pre-addestrato su Imagenet che abbiamo deciso di impiegare è VGG16 [SZ15]. Si tratta di un modello di CNN sviluppato seguendo la logica già introdotta da [KSH12]. Il modello impiegato difatti non linearità (ReLU) e uno stack di layer convoluzionali che fungono da *feature extractor* per la pipeline di classificatori finali. Abbiamo deciso di effettuare il taglio in tre punti:

- block5_conv1
- block4_conv
- block3_conv1

I tre punti corrispondono, nel modello, alle convoluzioni che seguono i 3 max pooling. Empiricamente, questi tre passaggi dovrebbero corrispondere a tre punti di in cui le feature risultino indipendenti e utili al task di **transfer learning**. Abbiamo estratto e visualizzato un subset delle activations per i tre punti, sovrapponendole all'immagine di input. Il risultato è riportato in figura 4.

3.2 ResNet50

Il secondo modello di CNN pre-addestrato su Imagenet che abbiamo deciso di impiegare è ResNet50. Si tratta di un modello di CNN che introduce il concetto di *residual*. Lo schema di ResNet in figura 5 è esplicativo: ciascun blocco di layer convoluzionali prevede uno *shortcut* dall'input all'output, ossia l'input del blocco viene sommato all'output. In questo modo, ResNet semplifica la fase di backpropagation del gradiente, che risulta essere più robusto anche con un numero elevato di layer. Abbiamo deciso di effettuare il taglio in quattro punti:

- conv2_block3_out
- conv3_block4_out

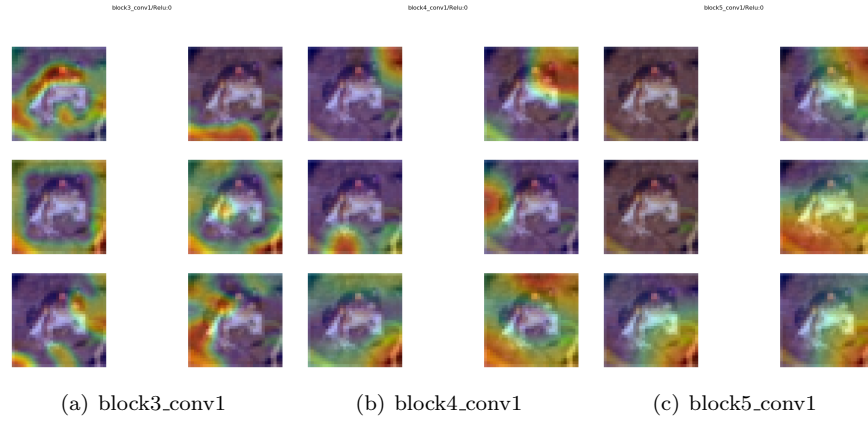


Figura 4: Heatmaps relative a un'immagine di input per le tre convoluzioni di taglio di VGG16

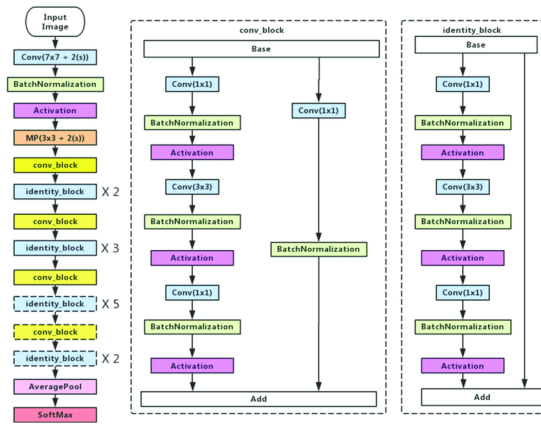


Figura 5: Resnet50

- conv4_block6.out
- conv5_block3.out

Anche in questo caso, i quattro punti corrispondono, nel modello, ai blocchi che seguono i max pooling. Abbiamo optato per tagliare la rete in due estremità e in due punti intermedi, al fine di valutare la bontà del *transfer learning* a differenti profondità. Riportiamo in figura 4 le heatmaps per un'immagine d'esempio.

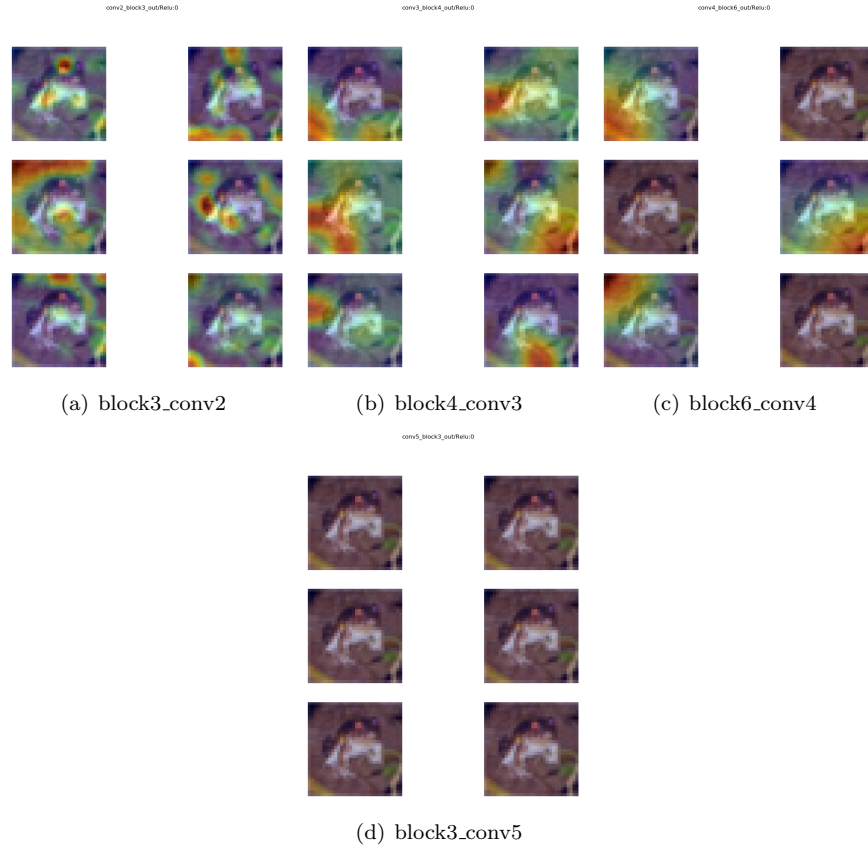


Figura 6: Heatmaps relative a un'immagine di input per le quattro convoluzioni di taglio di ResNet50

4 Esperimenti

Nella fase di sperimentazione abbiamo valutato diversi tagli impiegando un SVM lineare, per valutare la bontà dello spazio delle features a diversi livelli di taglio, senza introdurre non linearità. I parametri che abbiamo impiegato per il classificatore sono i seguenti:

- *kernel*: linear
- *penalty*: l2
- *loss*: squared_hinge
- *epochs*: 1000

Sono di seguito riportati i risultati ottenuti per il modello Linear SVM, divisi per CNN e livello di estrazione delle features¹.

4.1 Risultati ResNet50 - SVM

Riporto in figure 7 e 8 i risultati ottenuti dal modello di ML per i quattro livelli di taglio.

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.83	0.84	1000	5	0.88	0.86	1000
6	0.92	0.92	1000	6	0.92	0.94	1000
7	0.89	0.88	1000	7	0.90	0.90	1000
8	0.94	0.94	1000	8	0.94	0.95	1000
9	0.93	0.93	1000	9	0.94	0.93	1000
avg / total	0.90	0.90	5000	avg / total	0.92	0.92	5000

(a) conv2_block3_out

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.82	0.82	1000	5	0.80	0.70	1000
6	0.90	0.90	1000	6	0.84	0.84	1000
7	0.85	0.84	1000	7	0.74	0.80	1000
8	0.93	0.92	1000	8	0.84	0.87	1000
9	0.90	0.90	1000	9	0.82	0.83	1000
avg / total	0.88	0.88	5000	avg / total	0.81	0.81	5000

(b) block4_conv3

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.87	0.84	1022	5	0.88	0.87	1022
6	0.92	0.94	980	6	0.93	0.94	980
7	0.89	0.89	975	7	0.90	0.90	975
8	0.95	0.94	1001	8	0.96	0.95	1001
9	0.92	0.93	1022	9	0.94	0.94	1022
avg / total	0.91	0.91	5000	avg / total	0.92	0.92	5000

(c) block6_conv4

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.83	0.81	1022	5	0.73	0.73	1022
6	0.89	0.91	980	6	0.82	0.86	980
7	0.85	0.84	975	7	0.81	0.71	975
8	0.91	0.92	1001	8	0.86	0.83	1001
9	0.90	0.90	1022	9	0.79	0.87	1022
avg / total	0.87	0.87	5000	avg / total	0.80	0.80	5000

(d) block3_conv5

Figura 7: Risultati su *Test set* per i quattro livelli di taglio di ResNet50

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.87	0.84	1022	5	0.88	0.87	1022
6	0.92	0.94	980	6	0.93	0.94	980
7	0.89	0.89	975	7	0.90	0.90	975
8	0.95	0.94	1001	8	0.96	0.95	1001
9	0.92	0.93	1022	9	0.94	0.94	1022
avg / total	0.91	0.91	5000	avg / total	0.92	0.92	5000

(a) conv2_block3_out

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.83	0.81	1022	5	0.73	0.73	1022
6	0.89	0.91	980	6	0.82	0.86	980
7	0.85	0.84	975	7	0.81	0.71	975
8	0.91	0.92	1001	8	0.86	0.83	1001
9	0.90	0.90	1022	9	0.79	0.87	1022
avg / total	0.87	0.87	5000	avg / total	0.80	0.80	5000

(b) block4_conv3

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.87	0.84	1022	5	0.88	0.87	1022
6	0.92	0.94	980	6	0.93	0.94	980
7	0.89	0.89	975	7	0.90	0.90	975
8	0.95	0.94	1001	8	0.96	0.95	1001
9	0.92	0.93	1022	9	0.94	0.94	1022
avg / total	0.91	0.91	5000	avg / total	0.92	0.92	5000

(c) block6_conv4

precision	recall	f1-score	support	precision	recall	f1-score	support
5	0.83	0.81	1022	5	0.73	0.73	1022
6	0.89	0.91	980	6	0.82	0.86	980
7	0.85	0.84	975	7	0.81	0.71	975
8	0.91	0.92	1001	8	0.86	0.83	1001
9	0.90	0.90	1022	9	0.79	0.87	1022
avg / total	0.87	0.87	5000	avg / total	0.80	0.80	5000

(d) block3_conv5

Figura 8: Risultati su *Validation set* per i quattro livelli di taglio di ResNet50

4.2 Risultati VGG16 - SVM

Riporto in figura 9 e 10 i risultati su validation set e test set ottenuti dal modello di ML per i tre livelli di taglio del modello VGG.

¹Il framework impiegato per l'addestramento, scikit-learn, non permette l'accesso alla storia di training. Non dispongo pertanto di plot divisi per epoche

precision	recall	f1-score	support	precision	recall	f1-score	support		
5	0.71	0.76	0.73	1000	5	0.78	0.76	0.77	1022
6	0.84	0.85	0.84	1000	6	0.84	0.89	0.87	980
7	0.78	0.76	0.77	1000	7	0.83	0.77	0.80	975
8	0.87	0.86	0.86	1000	8	0.88	0.90	0.89	1001
9	0.85	0.83	0.84	1000	9	0.86	0.87	0.86	1022
accuracy			0.81	5000	accuracy			0.84	5000
weighted avg	0.81	0.81	0.81	5000	weighted avg	0.84	0.84	0.84	5000

(a) block3_conv1

(b) block4_conv1

precision	recall	f1-score	support	
5	0.79	0.75	0.77	1022
6	0.83	0.87	0.85	980
7	0.79	0.77	0.78	975
8	0.87	0.86	0.87	1001
9	0.81	0.85	0.83	1022
accuracy			0.82	5000
weighted avg	0.82	0.82	0.82	5000

(c) block5_conv1

Figura 9: Risultati su *Validation set* per i tre livelli di taglio di VGG16

5 Conclusione

Il task dell'assignment prevedeva di effettuare *transfer learning* di un modello preaddestrato su imagenet, impiegando il modello stesso come *feature extractor* per un dataset differente. Abbiamo deciso di impiegare Cifar10, ridotto a sole 5 classi, per valutare le feature estratte, essendo le classi di cifar a loro volta un subset di imagenet. I modelli di CNN che abbiamo impiegato sono VGG16 e ResNet50, i quali sono tagliati rispettivamente in 3 e 4 punti e quindi impiegati per la rappresentazione del dataset in features. L'approccio ha dato risultati generalmente migliori con il modello ResNet50, come si evince dal confronto tra le figure 8 e 10. In particolare, i tagli in un punto intermedi della rete, block4_conv3, ha prodotto un accuracy su test set del 91 %, contro il massimo valore raggiunto di 84 % per il taglio block4_conv1 del modello VGG16. La maggior profondità della ResNet50 ha sicuramente contribuito nell'estrarre features ben discriminanti. Dalle features maps in figura 6 e 4 si nota una maggior densità di features sul soggetto da classificare per i livelli di taglio che hanno portato a performance migliori.

	precision	recall	f1-score	support		precision	recall	f1-score	support
5	0.71	0.74	0.73	1022	5	0.76	0.79	0.78	1000
6	0.82	0.84	0.83	980	6	0.87	0.85	0.86	1000
7	0.78	0.76	0.77	975	7	0.82	0.80	0.81	1000
8	0.89	0.86	0.87	1001	8	0.85	0.90	0.88	1000
9	0.85	0.85	0.85	1022	9	0.88	0.84	0.86	1000
accuracy			0.81	5000	accuracy			0.84	5000
weighted avg	0.81	0.81	0.81	5000	weighted avg	0.84	0.84	0.84	5000

(a) block3_conv1

(b) block4_conv1

	precision	recall	f1-score	support
5	0.81	0.74	0.78	1000
6	0.85	0.89	0.87	1000
7	0.82	0.81	0.81	1000
8	0.86	0.89	0.88	1000
9	0.86	0.87	0.86	1000
accuracy			0.84	5000
weighted avg		0.84	0.84	5000

(c) block5_conv1

Figura 10: Risultati su *test set* per i tre livelli di taglio di VGG16

Bibliografia

- [Den+09] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*. 2009.
- [KSH12] Alex Krizhevsky, Ilya Sutskever e Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: (2012). A cura di F. Pereira et al., pp. 1097–1105. URL: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [SZ15] Karen Simonyan e Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations*. 2015.
- [KNH] Alex Krizhevsky, Vinod Nair e Geoffrey Hinton. “CIFAR-10 (Canadian Institute for Advanced Research)”. In: (). URL: <http://www.cs.toronto.edu/~kriz/cifar.html>.