Advanced Machine Learning: Assignment 3

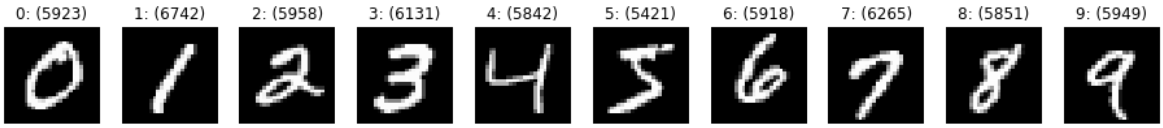
Davide Brinati

Matricola: 771458

1 Introduzione

L'obiettivo dell' assignment è quello di costruire una rete neurale, di tipo convoluzionale, da implementare per la classificazione di immagini. La dimensione di tali immagini è di 28x28 pixel. Esse compongono un dataset di 60.000 oggetti per il train e di 10.000 per il test, e rappresentano dei numeri scritti a mano.

Di seguito è possibile osservare alcuni elementi del train.



Verranno utilizzate due tipi di rete, e si opterà per quella che porterà a valori maggiori di accuracy sul test.

2 Modello 1

Il primo modello creato riceve dati in input di dimensione 28x28 ed è composto da un primo strato convoluzionale con 32 filtri, di dimensione 3x3 pixel e con stride pari a 2. Date queste dimensioni al kernel e allo stride, e tenendo presente che le immagini in input sono di dimensione 28x28 pixel, l'output di questo strato sarà di dimensione 13x13 pixel. La funzione di attivazione è la *relu*.

Successivamente è presente uno strato di *maxpooling*, con un pool size di 2x2. In questo strato l'informazione viene condensata, e dato che le finestre di pooling sono composte da 4 pixel, con stride uguale al pool size (ovvero 2x2) e che scorrono su un'immagine di 13x13, l'output di tale strato è composto da 6x6 pixel.

E' presente poi, un secondo strato convoluzionale, questa volta composto da 10 filtri con kernel size pari a 3x3. Lo stride risulta essere pari a 1, perciò le finestre convoluzionali di questo strato produrranno output di dimensione 4x4 pixel; la funzione di attivazione è relu. Il successivo strato di maxpooling condensa ulteriormente le informazioni, dando come output oggetti di dimensione 2x2.

Dopo gli strati convoluzionali è presente un flatten, necessario per appiattire i tensori che arrivano come input in vettori.

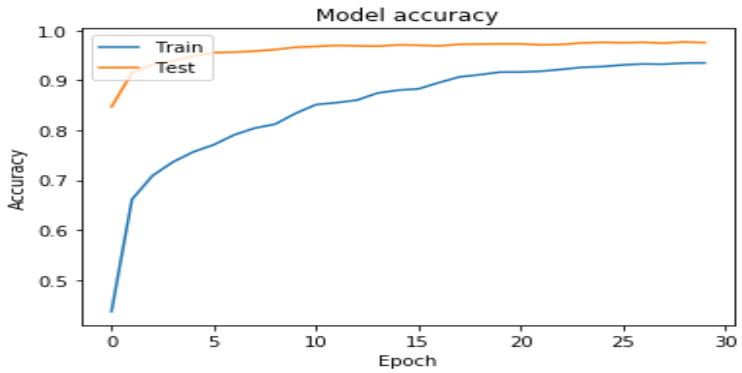
La parte successiva della rete è quella *fully connected*: è subito presente un layer Dense con 40 neuroni, seguito da un'altro strato Dense con 20 neuroni, entrambi con funzione di attivazione relu. Successivamente è presente un layer dropout, con un valore pari a 0.5, con il compito di eliminare il 50% delle connessioni fra lo strato precedente e quello successivo durante l'apprendimento, in modo da evitare il possibile overfitting. Infine abbiamo il layer di output, con 10 neuroni (tanti quante sono le possibili classi), e funzione di attivazione *softmax*.

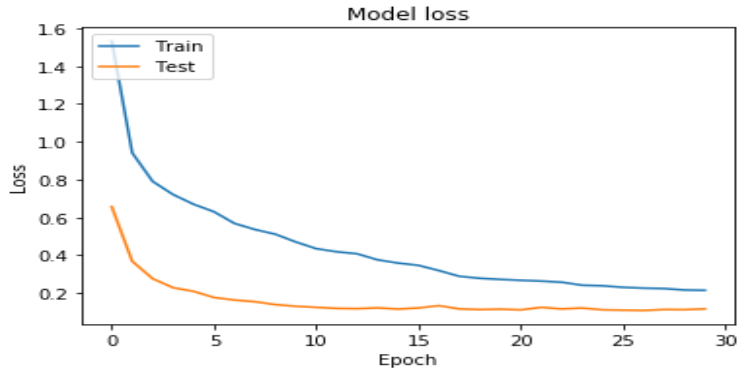
Ogni layer ha il seguente numero di parametri e produce un output delle seguenti dimensioni.

Layer (type)	Output Shape	Params Number
conv2d 1 (Conv2D)	(None, 13, 13, 32)	320
max pooling2d 1 (MaxPooling2d)	(None, 6, 6, 32)	0
conv2d 2 (Conv2D)	(None, 4, 4, 10)	2890
max pooling2d 2 (MaxPooling2d)	(None, 2, 2, 10)	0
flatten 1 (Flatten)	(None, 40)	0
dense 1 (Dense)	(None, 40)	1640
dense 2 (Dense)	(None, 20)	820
dropout 1 (Dropout)	(None, 20)	0
dense 3 (Dense)	(None, 10)	210
Total Params		5880

Il parametri sono 5880. La funzione di loss scelta è ovviamente la *categorical crossentropy*, l'ottimizzatore è adam, con il learning rate lasciato al suo valore di default ovvero 0.001; il batch size è stato impostato pari a 128

Il modello è stato addestrato per 30 epoche, producendo i seguenti risultati di accuracy e loss sul train e test:





3 Modello 2

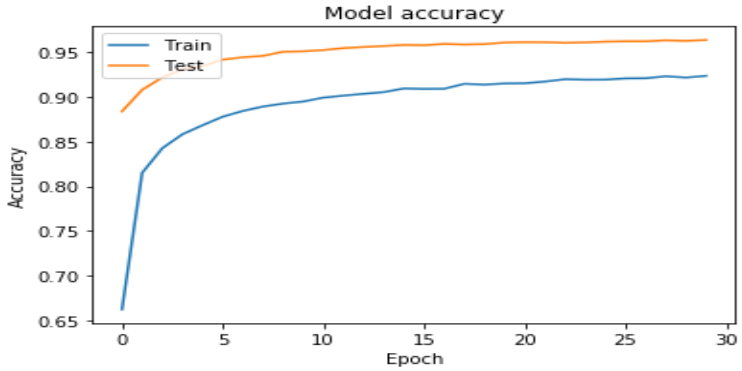
Il secondo modello utilizzato ha la seguente struttura: un solo layer convoluzionale, con 6 filtri di dimensione 3x3, con stride 2 e funzione di attivazione relu. Successivamente è presente un layer di maxpooling, con pool size pari a 2x2 pixel. Gli output di questi strati di convoluzione e pooling sono tensori, di dimensione 6x6 pixel su 6 canali.

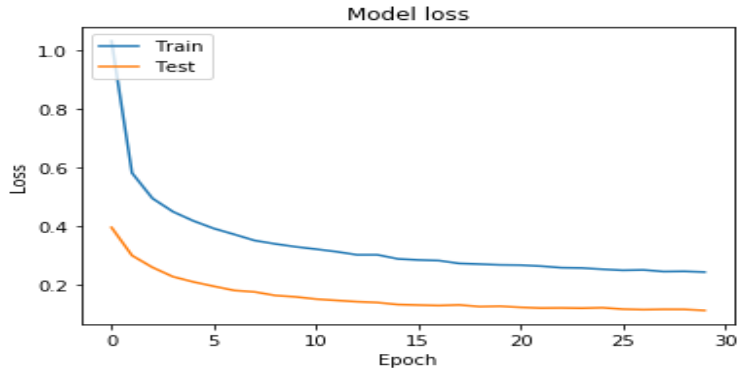
Prima della parte fully connected è presente un layer flatten, per appiattire i tensori che arrivano come input in vettori. Successivamente è presente un layer Dense con 40 neuroni, con funzione di attivazione

relu, seguito da un Dropout, e infine il layer di output, con 10 neuroni e funzione di attivazione softmax. Il numero di parametri per layer risulta essere:

Layer (type)	Output Shape	Params Number
conv2d 1 (Conv2D)	(None, 13, 13, 6)	60
max pooling2d 1 (MaxPooling2d)	(None, 6, 6, 6)	0
flatten (Flatten)	(None, 216)	0
dense 1 (Dense)	(None, 40)	8680
dropout 1 (Dropout)	(None, 40)	0
dense 2 (Dense)	(None, 10)	410
Total Params		9150

La funzione di loss è categorical crossentropy, l'optimizer adam con il learning rate lasciato al suo valore di default. L'apprendimento è stato effettuato con un valore del batch size pari a 128 e per 30 epoche.

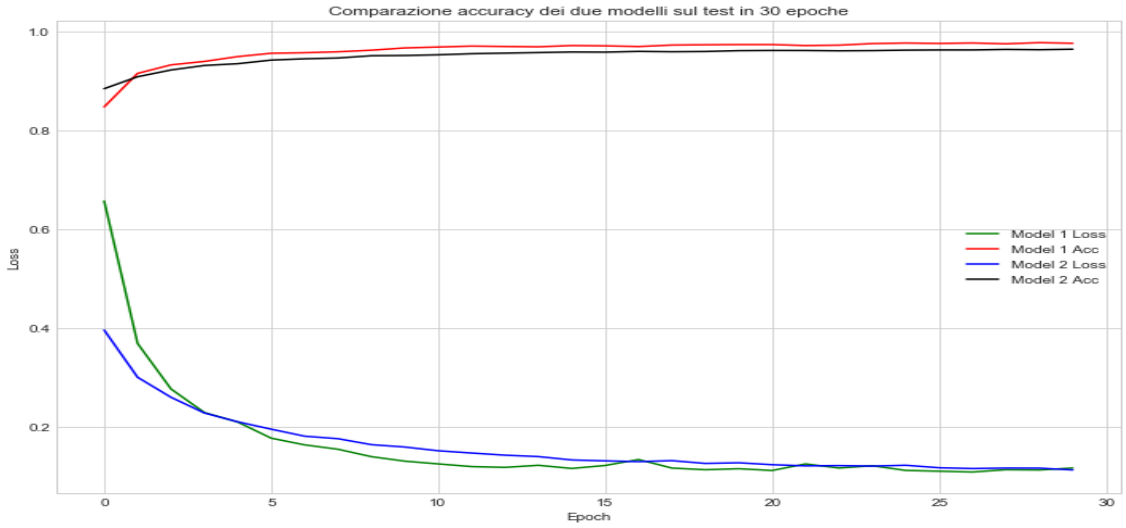




4 Conclusioni

Si effettua ora un confronto dell'andamento della curva di loss e dell'accuracy dei due modelli sul test, durante le 30 epoche.

L'accuracy sul test del primo modello (linea rossa) è risulta essere superiore rispetto a quella raggiunta dal secondo (linea nera). La loss del primo modello (linea verde) risulta essere inferiore rispetto a quella del secondo (linea blu) per quasi tutte le epoche del train. Per questi motivi verrà preferito il **primo modello**, il quale risulta avere anche meno parametri (5840) rispetto al secondo (9150).



Il primo modello raggiunge, sul test, un'**accuracy di 0.9757** e una **loss pari a 0.1163**. I valori di precision, recall e f1 di ciascuna classe sono i seguenti:

Classe	Precision	Recall	f1-score
0	0.98	0.99	0.98
1	0.99	0.99	0.99
2	0.97	0.98	0.98
3	0.99	0.97	0.98
4	0.98	0.98	0.98
5	0.95	0.98	0.96
6	0.97	0.98	0.98
7	0.97	0.96	0.96
8	0.98	0.97	0.97
9	0.98	0.95	0.96
total	0.98	0.98	0.98

Di seguito è possibile osservare tramite la confusion matrix, i risultati di classificazione sul test.