Assignment 2 (Advanced Machine Learning)

Davide Sangalli - Matricola: 848013

October 28, 2019

Abstract

In questo assignment vengono affrontati due problemi: la classificazione di immagini tramite reti neurali e la ricostruzione delle immagini stesse attraverso l'utilizzo di un autoencoder.

1 Data Preprocessing

1.1 Preparazione delle immagini

E' stato innanzitutto necessario effettuare qualche operazione di preprocessing alle immagini, partendo dal file *"x_train.obj"*, al fine di poterle utilizzare correttamente:

- è stata operata una conversione di tipo a *float_32* e sono state normalizzate dividendo ognuna per 255
- è stato effettuato un reshape delle immagini, da (28,28) a (784,1), ovvero si è passati da una rappresentazione in forma matriciale ad una rappresentazione di tipo monodimensionale
- non è stato necessario convertire la scala del colore delle immagini, poichè esse ci sono già state fornite in scala di grigi

1.2 Preparazione delle labels

Partendo dal file *"y_train.obj"*, è stato anche necessario effettuare una traslazione delle labels: esse avevano un range da 16 a 26, ma, per poterle utilizzare correttamente all'interno di Keras, sono state riportate all'interno di un range da 0 a 10 (quindi 11 classi totali). Le classi sono poi state trasformate tramite la codifica di *one hot encoding*.
Infine, è stato effettuato lo splitting del dataset in training set e validation set, usando il 20% dei dati per la validation ed il restante per il training. Inoltre è stato fissato un seme randomico di 42 per effettuare lo splitting.

2 Classificazione tramite rete neurale

Il primo approccio di classificazione è stato fatto con una rete neurale.

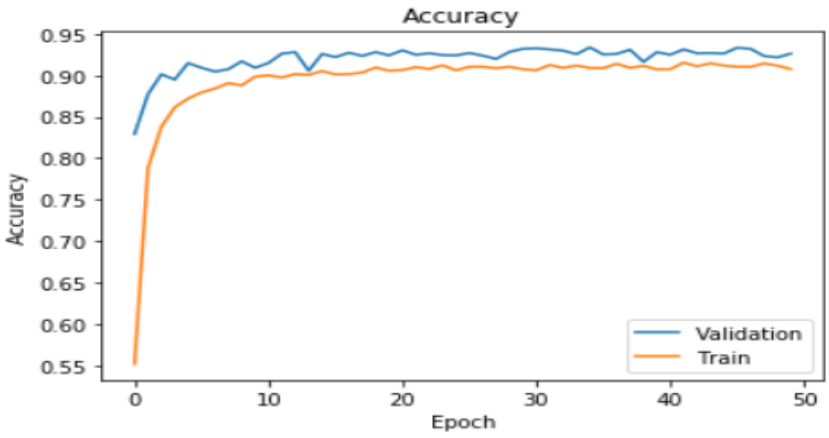
2.1 Architettura della rete

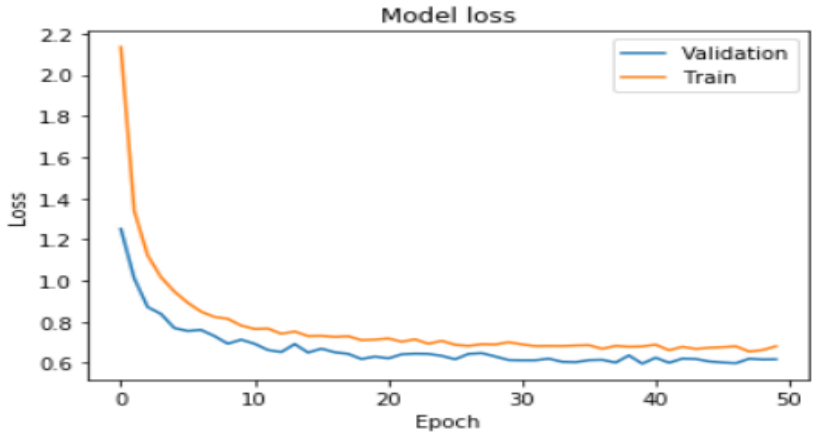
La rete neurale ha la seguente struttura:

- ha un layer di input, con *shape=(784,1)*
- tutti gli altri layers sono di tipo *Dense* (per i particolati vincoli imposti non è stato possibile utilizzare layers convoluzionali)
- ha un totale di 5 hidden layers di tipo *Dense*, così strutturati: il primo ha 512 neuroni (fattore di compressione dell'immagine di circa 1.5), il secondo ha 256 neuroni, il terzo 128, il quarto 64 ed infine l'ultimo ne ha 32. Dal secondo al quinto layer si è adottato un fattore di compressione pari a 2.
- il layer di output ha infine 11 neuroni, corrispondenti alle diverse classi di questo problema.
- tutti gli hidden layers hanno come funzione di attivazione la *relu*, ad eccezione dell'output layer che usa la funzione di attivazione *softmax*
- il modello è stato poi compilato usando *adam* come ottimizzatore, *categorical_crossentropy* come funzione di loss e usando l'*accuracy* come metrica di performance.
- è stato necessario aggiungere anche dei layers di *Dropout* tra un hidden layer e l'altro ed anche delle regolarizzazioni sui pesi per evitare l'*overfitting* della rete. In particolare sono stati inseriti, nell'ordine, i seguenti *dropout_rate*: 0.4, 0.3, 0.2, 0.2, 0.2 .
  Inoltre tutte le regolarizzazioni dei pesi sono state scelte pari a 0.001.
  Si è notato che, dando delle penalizzazioni maggiori ai pesi degli hidden layers della rete, l'accuracy scendeva all' 89% circa (sia sul training set che sul validation set) mentre usando delle regolarizzazioni di 0.001 ed inserendo layers di Dropout con i rate definiti precedentemente, le performance del modello in termini di accuracy miglioravano e contemporanemante si evitava l'overfitting.

2.2 Conclusioni

Di seguito vengono riportati i grafici dell'*accuracy* e della *loss* relativi sia al training set che al validation set per quanto riguarda questa rete neurale.
Come si può notare, questa rete non overfitta, mantenendo quasi costanti i valori di accuracy e di loss, dalla decima epoca in poi, su entrambi i set.
Questi grafici si sono ripresentati pressochè inalterati dopo diverse esecuzioni del modello.





Andando a confrontare i *classification reports* su entrambi i sets, si nota che l'accuracy del modello sul validation set è del 92-93% circa e sul training set 95-96% circa, perciò un leggero overfit effettivamente c'è, ma è comunque tollerabile.

3 Costruzione dell'autoencoder

Per questo task è stato anche costruito un autoencoder. Questo particolare modello di rete neurale è in grado di ricostruire le immagini che gli vengono fornite in input, apprendendone le features più importanti.

3.1 Architettura dell'autoencoder

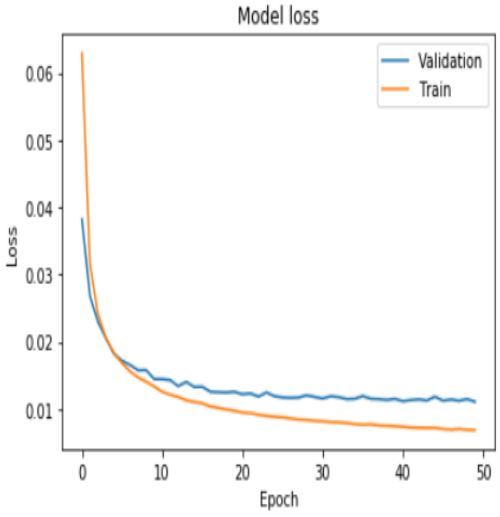
L'autoencoder ha la seguente struttura:

- un layer di input, con *shape=(784,1)*
- tre *Dense* layers di encoding che, nell'ordine, hanno il seguente numero di neuroni: il primo 512, il secondo 256 e il terzo 128. Come per la rete neurale, si è voluto applicare un fattore di compressione pari a 1.5 tra il layer di input e il primo hidden layer e per i restanti hidden layers si è mantenuto un rapporto di compressione pari a 2.
- il vero è proprio *code layer* è costituito da 64 neuroni
- i layers di decoding hanno una struttura speculare a quelli di encoding, perciò il layer più interno avrà 128 neuroni, quello successivo 256 ed infine l'ultimo 512.
- tutti gli hidden layers, tranne l'ultimo, hanno come funzione di attivazione la *relu*
- l'ultimo hidden layer ha come funzione di attivazione la *sigmoid*
- l'autoencoder è stato compilato utilizzando l'*adam* optimizer e come funzione di perdita l'*mse*

Dopo alcune prove, si è deciso di non applicare a nessun layer una regolarizzazione dei pesi e non è stato aggiunto nessun layer di Dropout, perchè l'introduzione di questi due elementi avrebbe portato a pessime performance dell'autoencoder, che non riusciva a ricostruire le immagini fornitegli in input.

3.2 Conclusioni

Il modello di encoder così costruito risulta particolarmente efficace: come mostrato sotto, il grafico della loss non presenta valori troppo discostanti per training e validation; inoltre si nota come la ricostruzione delle immagini da parte dell'encoder sia abbastanza accurata.



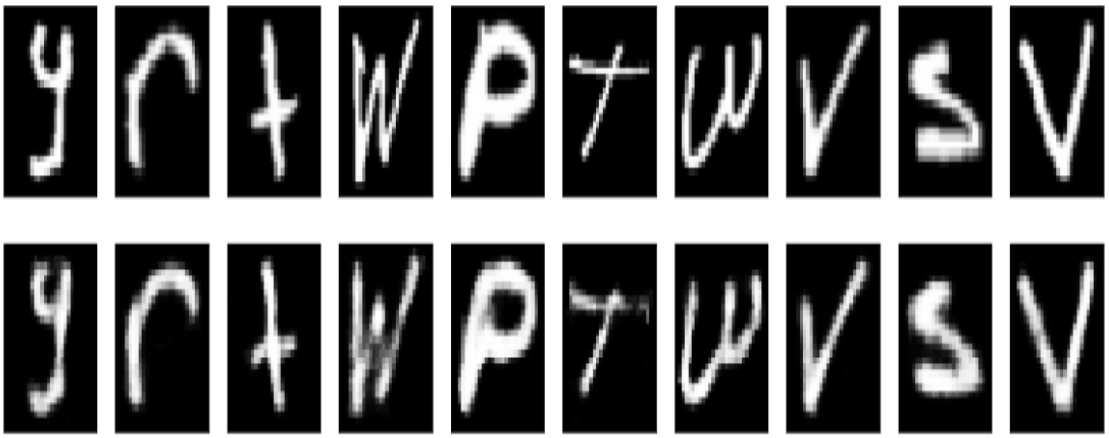


Figure 1: Ricostruzione delle immagini: la prima riga rappresenta le vere immagini del validation set, mentre la seconda riga rappresenta sempre le stesse immagini, ricostruite però dall'autoencoder

4 Costruzione della rete neurale partendo dall'autoencoder

Quest' ultimo approccio consiste nel costruire una rete neurale in cui una parte dei propri layers abbia la stessa struttura e gli stessi pesi dei layers dell'encoder.

4.1 Architettura della rete

Per quanto detto in precedenza, questo modello, dopo diversi tentativi, è stato così strutturato:

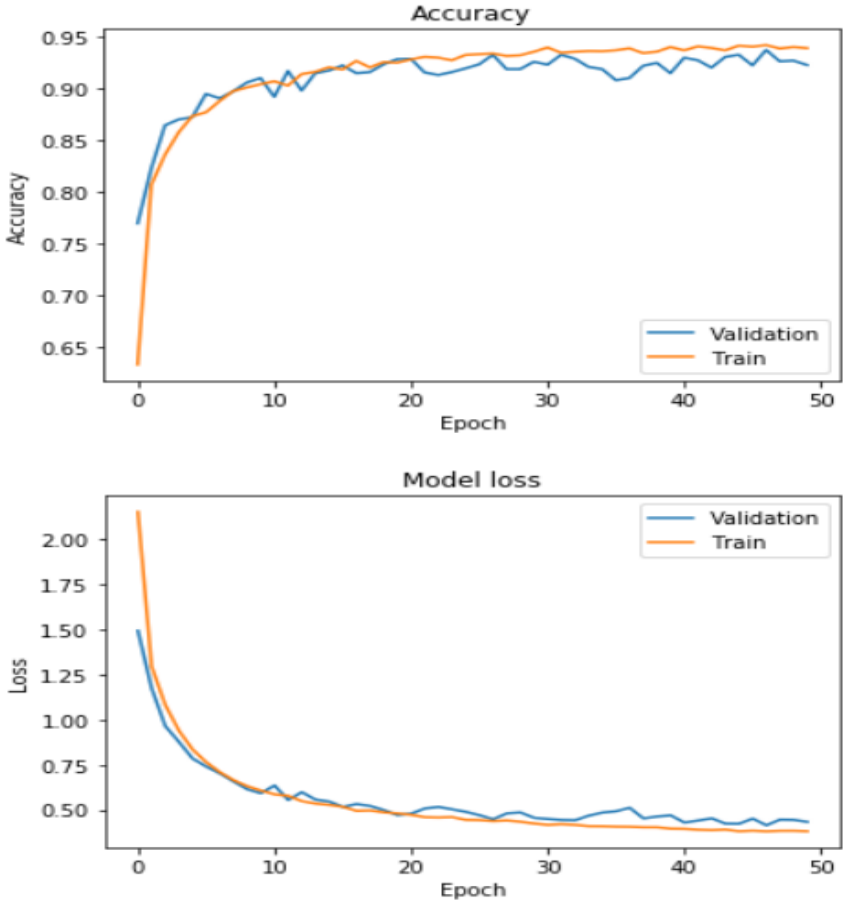
- un layer di input con *shape=(784,1)*
- quattro *Dense* layers (includendo anche il *code layer*) con la stessa architettura dei layers di encoding dell'autoencoder (vedi sopra per guardare la struttura). A questi layers sono stati dati gli stessi identici pesi dei layers dell'encoder definito precedentemente e il loro attributo *trainable* è stato settato a *False*, in modo da non essere allenati durante il training del modello.
- altri tre *Dense* layers aggiuntivi con, rispettivamente, 128, 64 e 32 neuroni. A questi tre layers sono state date delle penalizzazioni sui pesi (i primi

due layers hanno un *kernel_regularizer=0.01*, mentre per l'ultimo esso è pari a 0.001). Non è stato necessario aggiungere dei layers di Dropout.

- un layer di output con un numero di neuroni pari al numero di classi del problema in questione
- tutti gli hidden layers hanno come funzione di attivazione la *relu*, tranne il layer di output che ha come funzione di attivazione la *softmax*
- il modello, come fatto precedentemente per la rete neurale classica, è stato compilato usando l'*adam* optimizer, come funzione di loss la *categorical_crossentropy* e come metrica di performance l'*accuracy*

4.2 Conclusioni

L'architettura della rete neurale descritta sopra si è rivelata particolarmente efficace ai fini di una corretta classificazione delle immagini. Non c'è un sensibile overfitting, come si può notare dalla figura sottostante.



Confrontando poi i *classification reports* del training set e del validation set, si nota che c'è un overfitting minimo, comunque tollerabile; infatti l'accuratezza sul training set è di circa del 94-95% e quella sul validation set è di circa del 91-92%.
Si sono ripresentati dei risultati simili a questi anche con diverse esecuzioni del modello.
In conclusione, si può dire che l'accuratezza con cui vengono classificate le immagine usando questa architettura della rete neurale è pressochè simile a quella raggiunta per mezzo della rete neurale classica vista precedentemente.