Università degli Studi di Milano Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea magistrale in Informatica

ASSIGNEMENT 3
Advance Machine Learning

CNN for letters recognition

Andrea Guzzo, 761818

Descrizione dell'assignement

L'assignement consiste nella predizione di lettere rappresentate in scala di grigio dalla A alla J utilizzando una rete neurale convoluzionale che utilizzi almeno 10.000 parametri al suo interno.

Il Dataset contiene sia immagini di train che di test ottenute utilizzando il dataset di immagini MNIST

L'analisi compiuta è reperibile anche al seguente indirizzo su GitHub:

https://github.com/JeyDi/Digits-ConvNeuralNet

Soluzione

Prima della costruzione della rete neurale, è stato fatto un reshape del dataset MNIST utilizzato come training e test set.

È stata quindi costruita una Deep Neural Network Convoluzionale utilizzando la seguente struttura:

- Due layers convoluzionali 2D con una Relu come funzione di attivazione, con padding a 0 e stride = 3
- Due layers Max Pooling 2D con stride = 2
- Un layer di flattening che consente dopo la convoluzione e il pooling di convertire l'array multidimensionale ottenuto in un unico vettore lineare continuo utilizzabile con i successivi strati della rete.
- 1 Dense Layer con una funzione di attivazione RELU, con 16 parametri
- 1 Dense Layer con una funzione di attivazione SOFTMAX con 10 parametri che si occupa di effettuare la predizione finale delle 10 classi interessate.

Il modello quindi utilizza come funzione di ottimizzazione: rmsprop, che consente di dividere il tasso di apprendimento per un peso calcolato sulla media corrente delle grandezze dei gradienti recenti relativi a quel peso:

http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Come loss function viene utilizzata: categorical_crossentropy.

Per effettuare il fitting del modello sui dati, sono state usate 10 epoche con un early stop callback sulla funzione di loss che non entra mai in azione nei test effettuati. Mentre come batch size è stato scelto come valore: 254.

Il training della rete e l'evaluation è stato realizzato sulla CPU con un backend tensorflow.

Risultati

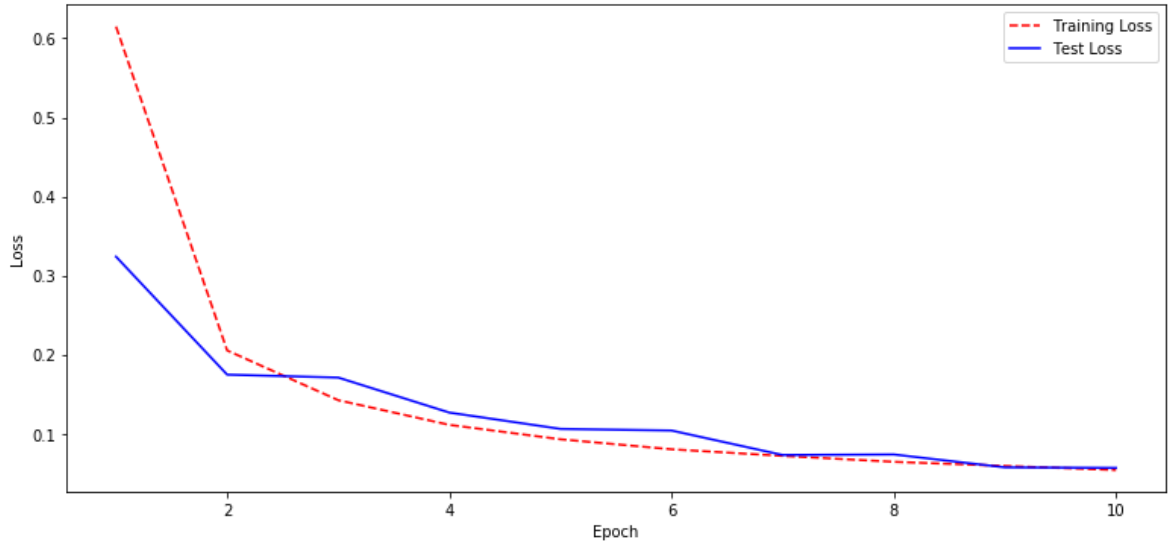
I risultati ottenuti effettuando il training della rete sono i seguenti:

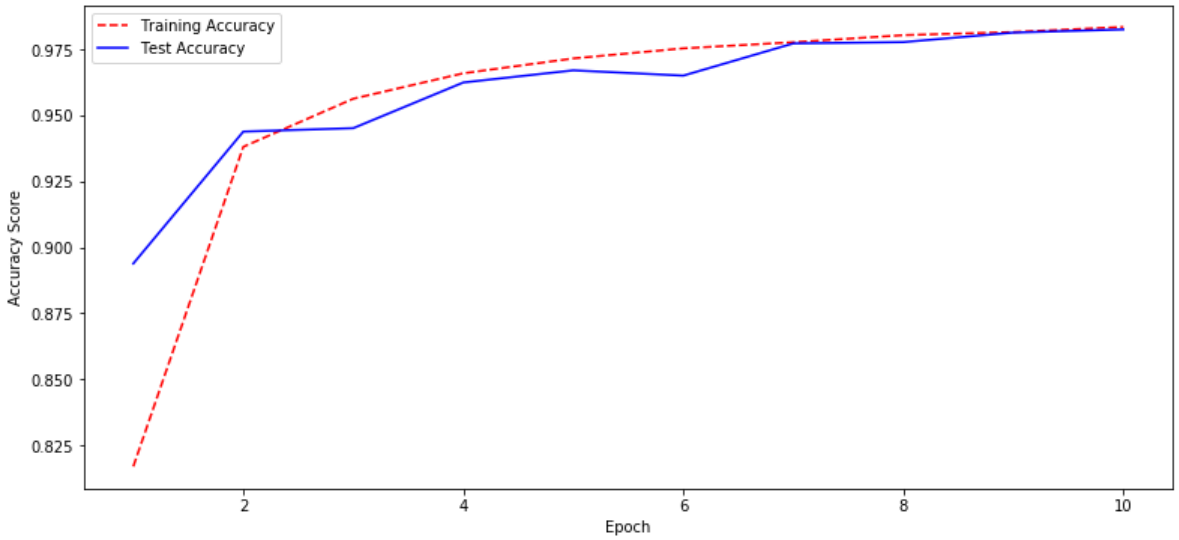
```
Training Accuracy 0.9834500078916549
Training Loss: 0.05444586953427642
Test Accuracy: 0.9825000083446502
Test Loss: 0.05715448258873075
```

La configurazione della rete rispetta i 10.000 parametri impostati nella consegna del progetto:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_10 (Conv2D)           (None, 26, 26, 16)        160
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 13, 13, 16)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 11, 11, 16)        2320
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 5, 5, 16)          0
_________________________________________________________________
flatten_5 (Flatten)          (None, 400)               0
_________________________________________________________________
dense_9 (Dense)              (None, 16)                6416
_________________________________________________________________
dense_10 (Dense)             (None, 10)                170
=================================================================
Total params: 9,066
Trainable params: 9,066
Non-trainable params: 0
```

I grafici ottenuti di accuratezza e di loss tra training e test hanno fornito i seguenti risultati:





Possiamo quindi dire di aver ottenuto dei buoni risultati con il dataset impiegato utilizzando la configurazione della rete presentata sulle macchine a disposizione.