Lorenzo Pastore
847212

Assignment 3

ADVANCED MACHINE LEARNING

Lo scopo di questo elaborato è quello di definire l'architettura ottimale di una rete neurale convoluzionale (CNN) finalizzata a risolvere un problema di classificazione non supervisionata sul dataset **MNIST Handwritten digits**.

1. Data exploration & Preprocessing

Il dataset consta di 60000 osservazioni di training e 10000 osservazioni di test costituite da input 28x28x1 relativi ad immagini di numeri scritti a mano. Le etichette del train set risultano essere bilanciate e si riferiscono ai numeri da 0 a 9.

Sono state effettuare alcune operazioni finalizzate a rendere i dati "suitable" per i modelli che sono stati sviluppati. In particolare sono state effettuare le seguenti operazioni:

- Train e test set sono stati convertiti in "float32" e normalizzati in una scala di grigi.
- Le etichette di classe del train set sono state convetite in una variabile categorica di 10 valori

In fine per favorire il confronto dei relativi modelli, il train set è stato diviso in due subset: Train set (80%) e Validation set (20%).

2. Modelling

Per la fase di modellazione è stato utilizzato Keras, inoltre il notebook è stato sviluppato in Google Colab con l'ausilio dell'hardware accelerator "GPU" per agevolare la valutazione della sensibilità dei parametri della CNN. Questa operazione rende tuttavia i risultati difficilmente riproducibili.

Nella seguente sezione vengono brevemente descritti i risultati delle analisi di sensibilità svolte sui parametri del modello. Ie quali sono alla base delle scelte effettuate in fase di definizione dell'architettura della rete convoluzionale finale.

Al fine di rendere la valutazione delle performance il più neutrale possibile è stato implementato sui modelli l'utilizzo di minibatch con dimensione 1024 per 10 epoche.

Quanti livelli di convoluzione utilizzare?

Sono stati costruiti tre modelli aventi da 1 a 3 layer di convoluzione e ne sono state valutate le performance in termini di accuracy. Come si può notare nella Fig.1 all'aumentare dei layer

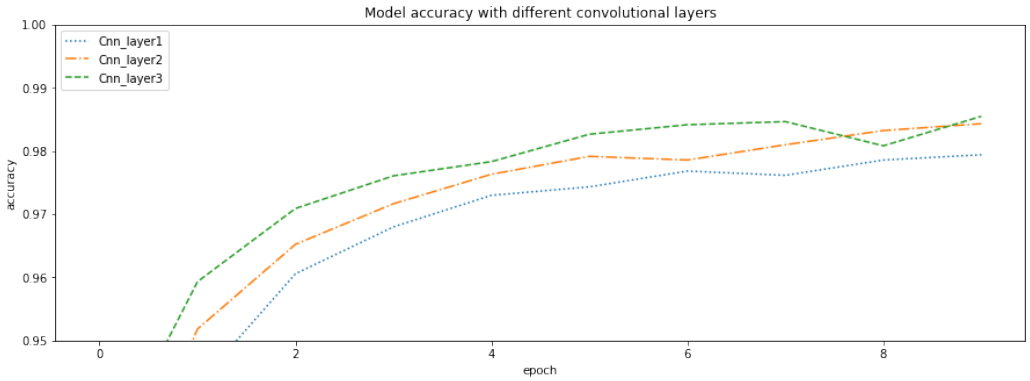


Fig. 1 - Model's accuracy with different n. of layers

aumenta la capacità predittiva del modello, tuttavia essendo il miglioramento tra il 2° e 3° modello molto modesto, s*i è deciso di utilizzare due layer di convoluzione.*

Quante features mappare?

In questo caso sono stati costruiti sei differenti modelli aventi due layer ciascuno sui quali si sono testate un numero di features pari ad 3 - 5 - 7 - 9 - 11 - 13 - 15 -17 nel primo layer e 4 - 6 - 8 - 10 - 12 - 14 - 16 -18 nel secondo layer.

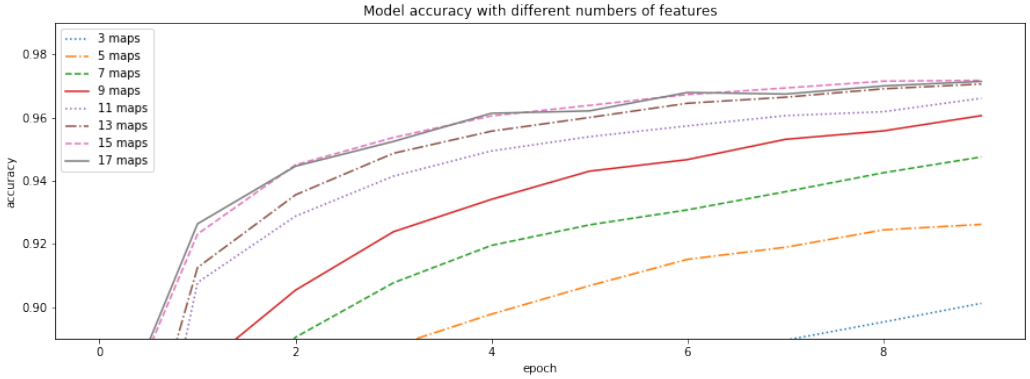


Fig. 2 - Model's accuracy with different n. of features

Anche in questo caso il vantaggio in termini di efficienza deve necessariamente tenere in considerazione il costo computazionale nonchè il vincolo imposto sul numero di parametri del modello, decidiamo quindi di utilizzare 11 features nel primo layer e 12 nel secondo.

Quanto grande deve essere il dense layer?

Sono stati costruiti otto modelli nei quali si sono testate le performance del modello andando a modificare il solo numero di unità presenti nel dense layer.

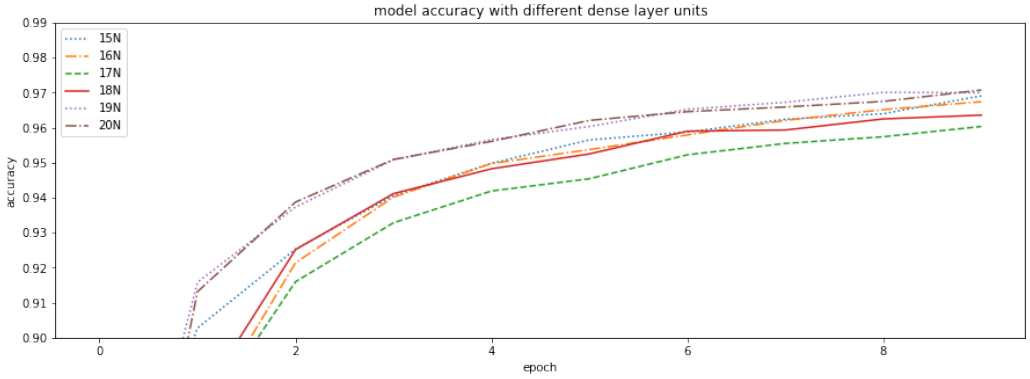


Fig. 3 - Model's accuracy with different dense units number

Dalla Fig.3 si evince come il modello con 17 unità abbia capacità predittive mediamente migliori di tutti gli altri modelli. *Si opta per un dense layer con 17 unità.*

Quanto dropout inserire nella rete?

Al fine di prevenire fenomeni di overfitting, decidiamo di effettuare alcuni test per capire quanto dropout inserire ad ogni layer.

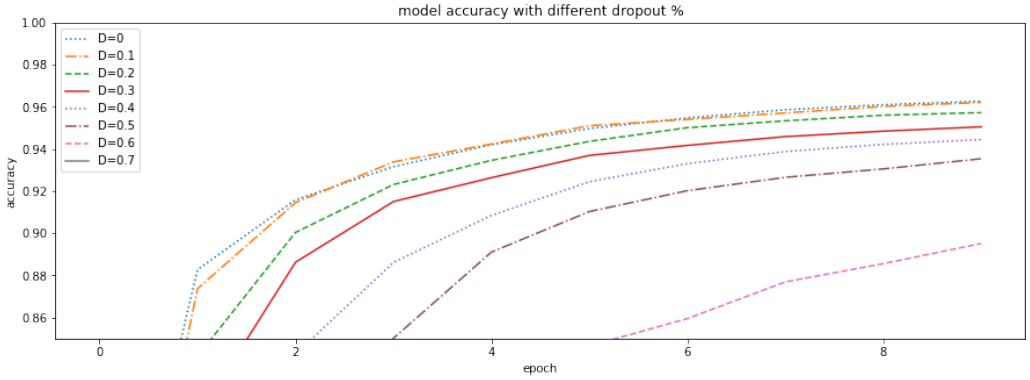


Fig. 4 - Model's accuracy with different dropout %

Dall'analisi è emerso che un *20% di dropout è il valore migliore per il modello sviluppato.*

Quanto deve essere grande il filtro?

Si è infine deciso di testare le performance di due modelli identici (n-layer, n-features, n-unità del dense layer) con due filtri di convoluzione differenti. Dai risultati è emerso che il modello con il filtro 5x5 raggiunge performance leggermente superiori rispetto al modello con filtro 3x3.

3. Final CNN architecture

Le scelte effettuate in fase di costruzione della rete sono in gran parte giustificate nella sezione precedente, di seguito si riporta una descrizione sintetica della rete neurale convoluzionale.

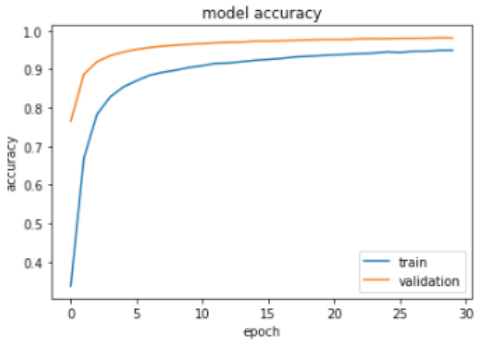
- Le immagini di **input** hanno una shape 28x28x1
- **Layer convoluzionale**: sono state mappate 11 features con un filtro 5x5 (stride=1) e funzione di attivazione "ReLu" che hanno prodotto un volume di output pari a (28-(5-1)) = 24x24x11. Il numero di parametri in questo livello è [*(5 x 5 x 1 + 1) x 11*] = 286
    - **Layer** di **max pooling** di dimensioni 2x2
    - **Layer di dropout** con dimensione 0.2 per prevenire l'overfitting e migliorare la generalizzazione
- **Secondo layer convoluzionale** con il quale sono state mappate 12 features con un filtro 5x5 e funzione di attivazione "ReLu" che ha prodotto un volume di outpur pari a (12-(5-1) = 8x8x12. Il numero di parametri in questo livello è pari a [*(5 x 5 x 11 + 1) x 12*] = 3312
    - **Layer** di **max pooling** di dimensioni 2x2
    - **Layer di dropout** con dimensione 0.2
- **Flatten layer** che genera un vettori di dimensione 192
- **Dense layer** costituito da 17 neuroni con funzione di attivazione "ReLu", il numero di parametri in questo livello è pari a [*(17 x 192)+17*] = 3281
    - **Layer di dropout** con dimensione 0.2
- **Output layer** costituito da 10 neuroni corrispondenti al numero di classi con funzione di attivazione "Softmax"

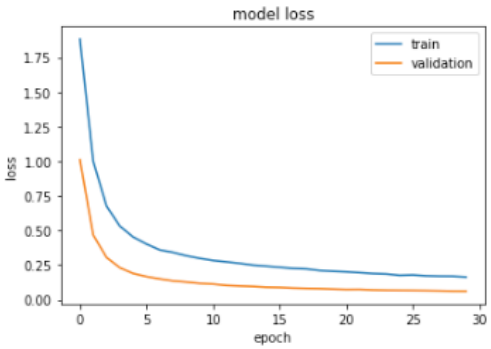
OPTIMIZATION FUNCTION AND HYPERPARAMETERS

E' stato utilizzato "Adam" con "categorical cross-entropy" come funzione di perdita valutata sul valore dell'accuracy del modello. L'algoritmo è stato allenato per 30 epoche su delle minibatch di 1024 unità.

4. Validation and Test set prediction

Il modello ha raggiunto un valore di accuracy di 0.9490 nel training e 0.9812 nel validation.





In conclusione il modello è stato utilizzato per predirre le etichette mancanti delle immagini contenute nel test set con un'accuracy di 0.9837 e un valore di loss pari a 0.0558.