Anselmo Raffaele
Matricola 846842
r.anselmo@campus.unimib.it

ASSIGNMENT 3

ADVANCED MACHINE LEARNING

L'assignment consiste nel design di una Rete Neurale Convoluzionale per risolvere il problema della classificazione supervisionata delle immagini contenenti cifre scritte a mano.

1. Dataset Overview and Preprocessing

Il dataset consiste di 60000 osservazioni di training e 10000 di test. Queste sono immagini delle cifre 0-9 scritte a mano. Le classi risultano abbastanza bilanciate.

Prima di procedere con la fase di learning, vengono eseguite delle operazioni sul dataset per renderlo di più facile computazione. Viene effettuato un *reshape* delle immagini in 4 dimensioni: la lunghezza del dataset, height (28), width (28) e depth (1), le immagini sono in scala di grigi. Successivamente normalizziamo i valori dividendoli per il valore massimo RGB, 255.

Il training set viene a sua volta suddiviso in train set (80%) e validation set (20%) per effettuare la validazione del modello.

Ai fini di valutazione dei modelli viene definita la funzione ***plot_metrics***.

2. Convolutional Neural Network Model

Con la funzione ***make_model*** si definisce la rete neurale convoluzionale con due layer convoluzionali e due di pooling seguiti da una rete fully-connected:

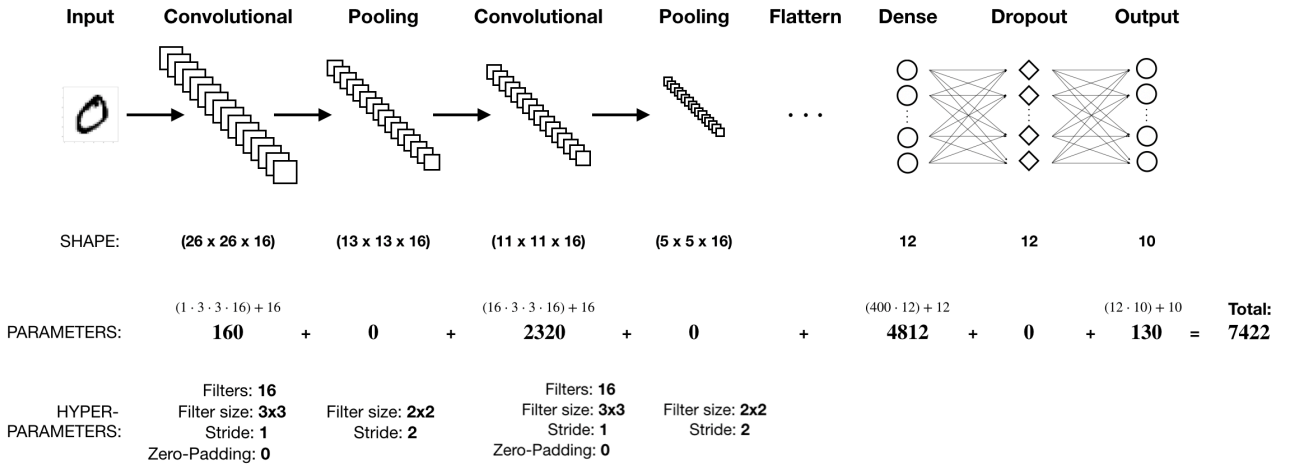

Fig. 1 CNN Architecture, Parameters and Hyperparameters

Nel primo Convolutional Layer sono stati applicati 16 filtri ai dati di input (di dimensione *28* x 28 x *1)* con *Receptive Field 3* x *3* e *stride* pari a 1 e funzione di attivazione *'ReLU'*. Questo produce un volume di output con dimensione *26* x *26* x *16.*

Caratteristica della rete neurale convoluzionale è l'asimmetria con cui tratta le dimensioni: i filtri agiscono localmente su altezza e larghezza del volume di input, ma lungo tutta la sua profondità.

Il Pooling layer implementa invece una funzione fissa non parametrica che diminuisce le dimensioni in termini di altezza e larghezza dell'Input. In particolare è stato utilizzato il *MaxPooling* che ridimensiona spazialmente ogni depth slice dell'input utilizzando l'operazione MAX con estensione 2x2 e passo (stride) 2.

A questo seguono un altro layer convoluzionale e un altro di pooling caratterizzati dalla stessa architettura dei due precedenti. Il volume di output ha dimensione *5x5x16* e, attraverso il *Flattern layer,* viene restituito un vettore di dimensione 400 ai Fully-Connected layers.

Il primo *Dense Layer* ha 12 neuroni, funzione di attivazione '*ReLU'* ed è succeduto da un layer di *Dropout* con *rate = 0.2*.

Il layer di output ha 10 neuroni, in modo da restituire le probabilità di appartenenza alle classi e ha funzione di attivazione *'softmax'*

Nel design della rete neurale si è tenuto conto del limite massimo di parametri (pesi + distorsioni) pari a 7500. Si noti come, malgrado i layer convoluzionali hanno dimensione maggiore rispetto a quelli fully-connected, il numero di parametri da stimare al loro interno sia molto più basso. Questo è dovuto al principio del *parameter sharing:* tutti i neuroni in una *depth slice* (la porzione bidimensionale lungo la profondità del volume, uguale al numero di filtri) condividono gli stessi pesi e le stesse distorsioni.
In particolare, il numero di parametri nel primo layer convoluzionale è dato da:

$$Input\ Depth \times Filter_{Height} \times Filter_{Width} \times Depth\ Slices + Biases$$
$$(1 \times 3 \times 3 \times 16) + 16 = 160$$

Mentre nel secondo layer convoluzionale è pari a:

$$(16 \times 3 \times 3 \times 16) + 16 = 2320$$

La maggior parte di parametri del modello risiede infatti all'interno del Fully-Connected Layer (*4812* nel primo *Dense Layer* e *130* nell'*Output Layer*). Questi possono essere controllati agendo sugli iperparametri del pooling layer, i quali determinano la diminuzione delle dimensioni dei volumi di output e, di conseguenza, il numero di parametri nel layer FC.

Come ottimizzatore è stata utilizzato *'Adam'* con funzione di perdita *'categorical crossentropy'.*
Per valutare le performances si utilizza l'*accuracy*.

Dopo aver definito gli ultimi iperparametri:

- *Learning rate = 0.001*
- *Epochs = 10*
- *Batch Size = 1024*

si passa alla fase di training del modello.

La rete neurale convoluzionale apprende molto bene e dai risultati sul validation set non solo si evince l'assenza di overfitting, ma si vede anche un ottima generalizzazione grazie all'effetto dei layers di Pooling e Dropout:

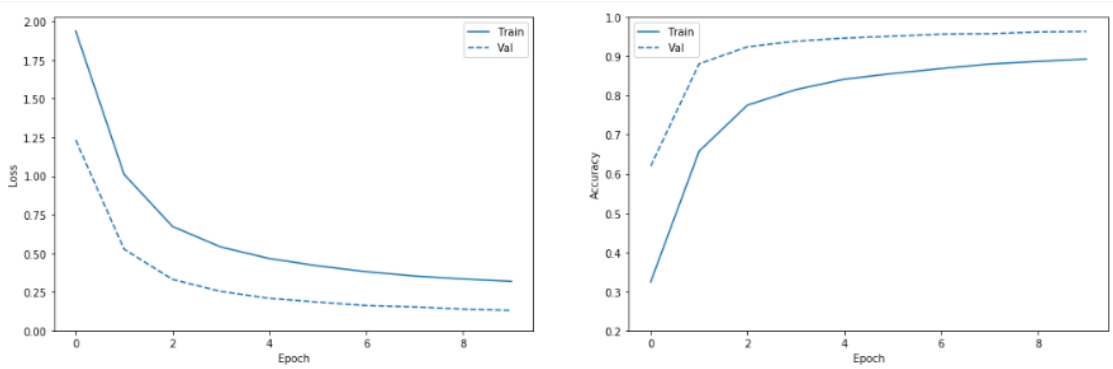

Fig. 2 Performance on Training set and Validation set

Si procede quindi con l'applicazione dell'algoritmo sul Test Set nel quale si ottiene un ***Accuracy*** pari a ***0.968*** e una ***Loss*** di ***0.115***.