

# ADVANCED MACHINE LEARNING

## REPORT ASSIGNMENT #3

Giugliano Mirko

matricola: 800226

### INTRODUZIONE

Il terzo assignment per il corso di “Advanced Machine Learning” consiste nella classificazione di numeri dallo 0 al 9, sviluppando una rete neurale convoluzionale, a partire dal dataset MNIST. Inoltre vi era tenere conto del numero di parametri utilizzati dalla rete per non superare il limite massimo di 7500.

### ANALISI ESPLORATIVA e PREPROCESSING

Durante l’esplorazione del dataset si nota subito che la dimensione del train set è di 60000 record, mentre di 10000 per il test set, ognuno contenente l’immagine di un numero da 0 a 9 in una matrice 28x28, che sta ad indicare la scala di grigi di ogni pixel dell’immagine. La normalizzazione per questo motivo è stata effettuata semplicemente dividendo ogni numero per 255, dato che il range dei numeri delle suddette matrici va da 0 a 255. Dopodichè viene effettuato un reshape, poiché i layer convoluzionali necessitano di sapere quanti canali vi sono per ogni immagine, in questo caso uno trattandosi di foto in bianco e nero. Quindi si è proceduto con la categorizzazione della variabile da classificare, tramite one hot encoding si sono ottenute 10 variabili di comodo necessarie per il problema.

### MODELLI

#### RETE NEURALE:

- 50 epoche
- 2 layer convoluzionali, con 10 filtri, di dimensione 3x3, no padding
- 2 pooling layer (MaxPooling2D) di dimensione 2x2, padding “same”
- 1 layers dense da 44 neuroni
- 2 dropout layer
- activation function layers: ReLU
- activation function output layer: Softmax
- optimizer: Adam
- batch size:252
- loss function: categorical\_crossentropy
- regularization: 3 dropout layers

Il numero di epoche è stato scelto in base a quando l’accuracy del train set e del test set arrivavano a convergenza, evitando overfitting e/o underfitting. Rispettando il limite imposto sui parametri, sono state effettuate le scelte sul numero di neuroni e sulla dimensione dei filtri nei layer convoluzionali. L’activation function ReLU è una scelta abbastanza standard e risultava la più performante mentre per l’activation function dell’output layer la Softmax è stata una scelta obbligata dal fatto che fosse una classificazione multiclasse. L’optimizer utilizzato per questo modello è stato RMSprop, algoritmo che risulta essere molto

utilizzato in letteratura quando si sviluppano reti neurali convoluzionali e che risultava infatti il più performante. Per evitare l'overfitting inoltre si sono introdotti due dropout layer.

## VALUTAZIONE MODELLO E CONCLUSIONI

### SUMMARY DEL MODELLO PER VERIFICARE CHE SIANO RISPETTATI I VINCOLI SUI PARAMETRI

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 26, 26, 10)	100
max_pooling2d_9 (MaxPooling2D)	(None, 13, 13, 10)	0
batch_normalization_9 (Batch Normalization)	(None, 13, 13, 10)	40
dropout_13 (Dropout)	(None, 13, 13, 10)	0
conv2d_10 (Conv2D)	(None, 11, 11, 4)	364
max_pooling2d_10 (MaxPooling2D)	(None, 6, 6, 4)	0
batch_normalization_10 (Batch Normalization)	(None, 6, 6, 4)	16
dropout_14 (Dropout)	(None, 6, 6, 4)	0
flatten_5 (Flatten)	(None, 144)	0
dense_9 (Dense)	(None, 44)	6380
dense_10 (Dense)	(None, 10)	450
Total params: 7,350		
Trainable params: 7,322		
Non-trainable params: 28		

### VALORI DELLA LOSS FUNCTION E DELLA ACCURACY SU TRAIN SET, VALIDATION SET E TEST SET

#### Train

- accuracy: 0.9877
- loss: 0.0369

#### Validation

- accuracy: 0.9878
- loss: 0.0397

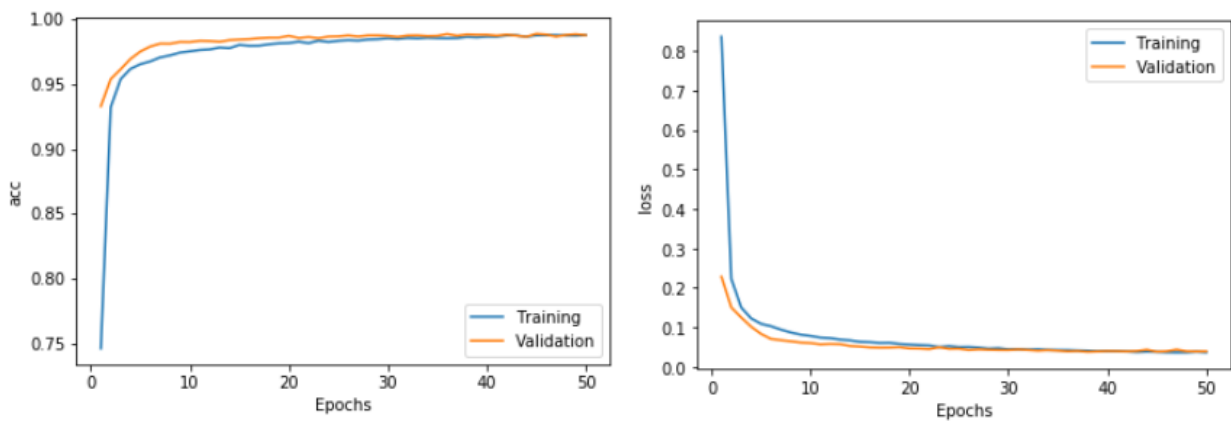
#### Test

- accuracy: 0.0381
- loss: 0.9880

### CONFUSION MATRIX E CLASSIFICATION REPORT

confusion matrix															
[[	975	0	0	0	0	0	4	1	0	0]					
[	0	1132	1	0	0	1	1	0	0	0]	0	0.99	0.99	0.99	980
[	2	1	1021	0	2	0	0	4	2	0]	1	0.99	1.00	0.99	1135
[	0	1	2	997	0	6	1	2	1	0]	2	0.99	0.99	0.99	1032
[	0	0	0	0	978	0	1	0	1	2]	3	0.99	0.99	0.99	1010
[	1	0	0	5	0	881	2	1	1	1]	4	0.99	1.00	0.99	982
[	4	2	0	0	3	3	945	0	1	0]	5	0.98	0.99	0.98	892
[	0	4	7	0	0	0	0	1014	1	2]	6	0.99	0.99	0.99	958
[	1	0	4	3	0	3	2	4	951	6]	7	0.98	0.99	0.98	1028
[	0	1	1	1	9	3	0	6	2	986]]	8	0.99	0.98	0.98	974
											9	0.99	0.98	0.98	1009
											accuracy			0.99	10000
											macro avg	0.99	0.99	0.99	10000
											weighted avg	0.99	0.99	0.99	10000

## ANDAMENTO ACCURACY E LOSS FUNCTION



### COMMENTO:

Le performance del modello sono ottime pur utilizzando pochi parametri, il rapporto fra complessità del modello e risultati, utilizzando questo tipo di rete per questo problema, è veramente apprezzabile.