ADVANCED MACHINE LEARNING

REPORT ASSIGNEMENT #2

Giugliano Mirko

matricola: 800226

INTRODUZIONE

Il secondo assignement per il corso di "Advanced Machine Learning" consiste nella classificazione di lettere dalla p alla z, sviluppando una rete neurale, a partire da un dataset contenente immagini delle suddette lettere. Inoltre era necessario sviluppare un autoencoder, per estrarre le caratteristiche più importanti dai dati di allenamento.

ANALISI ESPLORATIVA e PREPROCESSING

Durante l'esplorazione del dataset si nota subito che la dimensione del train set è di 14000 record, ognuno contenente l'immagine di una lettera dalla p alla z in una matrice 28x28, che sta ad indicare la scala di grigi di ogni pixel dell'immagine. La normalizzazione per questo motivo è stata effettuate 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 portando il dataset ad avere un singolo vettore di valori per ciascun campione, questo perchè il modello richiede dati in input in due dimensioni. Quindi si è proceduto con la categorizzazione della variabile da classificare, che, con un numero da 16 a 26, rappresentavano la lettera dell'alfabeto corrispondente a quella posizione. Tramite one hot encoding di sono ottenute 11 variabili di comodo utili per il problema. Si è deciso infine di splittare il train set per ottenere un 25% di record necessari per il validation set.

MODELLI

RETE NEURALE:

- 400 epoche
- 2 hidden layers in ordine da 512 e 256 neuroni.
- activation function hidden layers: ReLU
- activation function output layer: Softmax
- optimizer: Adam
- loss function: categorical_crossentropy
- regularization: 2 dropout layers e regolarizzatori L1 e L2

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. Il numero di layer e di neuroni è stato scelto in seguito a svariati tentativi, cercando un buon compromesso fra complessità e prestazioni. Per l'activation function degli hidden layers, la ReLU, quella comunamente più usata, risultava la più performante mentre per l'activation function degli output layer la Softmax è stata una scelta obbligata dal fatto che fosse una classificazione multiclasse. L'optimizer utilizzato per questo modello è stato Adam, algoritmo adattivo e molto utilizzato viste le sue prestazioni. Per evitatare l'overfitting si sono introdotti anche dei regolarizzatori L1 e L2 e due dropout layer.La loss function categorical_crossentropy è praticamente obbligatoria trattandosi di più classi da stimare.

AUTOENCODER

- 100 epoche
- Input layer da 784
- Encoder da 512 neuroni
- Hidden layer da 32 neuroni.
- Decoder da 512 neuroni
- Output layer da 784
- activation function ReLU, activation function output layer: Sigmoid
- optimizer: Adam
- loss function: binary_crossentropy

Anche per l'autoencoder i parametri sono stati scelti dopo svariati tentativi e compiendo scelte abbastanza standard in ambito reti neurali e giustificandoli come precedentemente fatto. La loss function è ovviamente diversa poiché essendo i pixel da ricostruire in un range fra 0 e 1, diventa necessaria avere una loss function che tenga conto del fatto che siano valori binari.

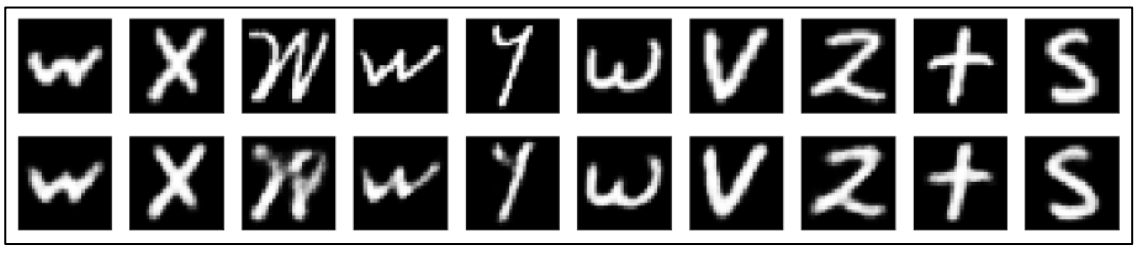
VALUTAZIONE MODELLI E CONCLUSIONI

PRIMO MODELLO:
confusion matrix e classification report

```
confusion matrix
 [[317  17   4   0   2   4   0   1   0   5   0]
 [  4 298   3   2   2   1   1   0   0   9   0]
 [  6   2 293   0   1   3   5   0   5   5   1]
 [  0   9   0 317   2   1   0   1   4   2   0]
 [  0   1  19   0 314   0   3   0   3   8   1]
 [  0   3   1   0   0 300  14   3   0   3   0]
 [  0   1   4   0   0  18 274   2   1   8   0]
 [  0   0   1   0   0   8   8 292   0   1   0]
 [  0   3   4   0   2   1   3   2 295   7   1]
 [  0   5   4   0   5   5   9   0   3 310   1]
 [  1   3   1   1   6   4   1   0   0   1 204]]
```

	precision	recall	f1-score	support
0	0.97	0.91	0.94	350
1	0.87	0.93	0.90	320
2	0.88	0.91	0.89	321
3	0.99	0.94	0.97	336
4	0.94	0.90	0.92	349
5	0.87	0.93	0.90	324
6	0.86	0.89	0.88	308
7	0.97	0.94	0.96	310
8	0.95	0.93	0.94	318
9	0.86	0.91	0.88	342
10	0.98	0.92	0.95	222
accuracy			0.92	3500
macro avg	0.92	0.92	0.92	3500
weighted avg	0.92	0.92	0.92	3500

AUTOENCODER:



Di seguito per valutare le performance di classificazione del modello si è sostituito l'ultimo layer di dimensione 784 con uno composto da 11 neuroni, ovvero il numero delle classi che possono rappresentare l'output. La funzione di attivazione viene cambiata e utilizzata la softmax in virtù delle più classi. I risultati però non migliorano rispetto al modello precedente.

	precision	recall	f1-score	support
0	0.96	0.92	0.94	339
1	0.93	0.89	0.91	335
2	0.91	0.92	0.91	337
3	0.97	0.96	0.97	329
4	0.93	0.90	0.91	327
5	0.87	0.91	0.89	320
6	0.84	0.92	0.88	317
7	0.90	0.95	0.93	310
8	0.92	0.93	0.92	347
9	0.92	0.84	0.88	337
10	0.94	0.93	0.93	202
accuracy			0.92	3500
macro avg	0.92	0.92	0.92	3500
weighted avg	0.92	0.92	0.92	3500

confusion matrix

```
[[312   9   6   0   2   1   2   4   0   3   0]
 [  4 297   4   4   3   8   1   5   3   5   1]
 [  6   3 310   0   1   3   4   2   5   1   2]
 [  0   1   3 317   2   1   0   1   2   1   1]
 [  1   3  11   1 294   0   1   0   6   5   5]
 [  0   1   0   1   0 292  16   7   3   0   0]
 [  0   1   2   0   1  14 292   4   0   2   1]
 [  0   0   1   0   0   5   6 296   2   0   0]
 [  0   0   1   0   2   2   5   4 322   9   2]
 [  0   3   3   1   7   7  21   3   7 284   1]
 [  1   0   1   2   4   2   0   3   1   0 188]]
```

COMMENTO:

Le performance del modello sono tendenzialmente buone e andando anche ad ispezionare visivamente gli errori di classificazione, alcuni anche ad occhio umano sarebbero difficilmente classificabili. L'autoencoder sul train set e sul validation set raggiunge un valore della loss function di circa 0.14 e anche visivamente sembra ricostruire bene le lettere estraendo le caratteristiche principali.