Machine Learning **Matricola: _____**

13-Set-2021 (90 minuti) **Cognome: _____ Nome: _____**

1) Quali sono i parametri di una distribuzione multinormale?

Dispense “Classificazione (1)”

2) Descrivere a grandi linee un classificatore Random Forest

Dispense “Classificazione (2)”

3) Indicare le differenze tra le tecniche di riduzione di dimensionalità PCA e LDA.

Dispense “Riduzione Dimensionalità”

4) Descrivere a grandi linee l’algoritmo di Clustering K-means.

Dispense “Clustering”

---

5) In una rete CNN, data un’immagine di Input di dimensione $7 \times 7 \times 3$ (nel formato $Width \times Height \times Depth$) e un livello di convoluzione composto da 1 filtro di dimensioni $3 \times 3 \times 3$ con $padding = 0$ e $stride = 2$, si calcoli il valore dell’elemento del volume di output indicato con il ?.

Input

Depth 0

197	103	42	252	27	78	205
114	57	2	195	7	1	130
97	71	179	60	187	22	21
86	84	187	229	208	167	237
25	177	236	250	25	9	87
217	175	190	175	23	10	69
67	127	246	142	4	125	87

Depth 1

124	164	158	18	229	152	110
19	111	22	75	167	224	88
136	21	201	237	248	43	136
151	245	140	163	12	207	19
212	197	87	203	42	149	157
12	78	232	52	113	232	198
64	167	99	112	42	236	186

Depth 2

105	45	160	12	81	207	228
174	18	111	216	200	91	62
170	191	128	124	74	187	123
199	224	184	134	66	193	87
77	41	50	226	226	88	106
151	182	191	216	198	184	93
60	120	91	168	141	136	150

Filtro

Depth 0

0.89	0.87	0
0	0.3	0.52
0	0	0.38

Depth 1

0.24	0.9	0
0.07	0	0.64
0.01	0	0.41

Depth 2

0	0.71	0
0.4	0	0
0	0.1	0.3

Output

-	-	-
-	-	-
-	?	-

Svolgimento

Nella convoluzione 3D, per ogni elemento del volume di output, il filtro opera su una porzione diversa del volume di input. Tale posizione dipende dai parametri $padding$ e $stride$ oltre che dalla posizione dell’elemento che si vuole calcolare (nel volume di output). La regione bordata di nero nell’immagine di Input rappresenta la porzione da considerare per calcolare l’elemento di output evidenziato.

Il valore della cella di output viene calcolato come somma dei prodotti di ogni elemento della porzione di input per l'elemento corrispondente del filtro.
Nell'ambito delle CNN, la convoluzione non richiede nessuna operazione di "ribaltamento" del filtro e di normalizzazione del risultato.

$\text{Depth } 0 = 236 \cdot 0.89 + 250 \cdot 0.87 + 175 \cdot 0.3 + 23 \cdot 0.52 + 4 \cdot 0.38 = 493.52$

$\text{Depth } 1 = 87 \cdot 0.24 + 203 \cdot 0.9 + 232 \cdot 0.07 + 113 \cdot 0.64 + 99 \cdot 0.01 + 42 \cdot 0.41 = 310.35$

$\text{Depth } 2 = 226 \cdot 0.71 + 191 \cdot 0.4 + 168 \cdot 0.1 + 141 \cdot 0.3 = 295.96$

$\text{Risultato} = 493.52 + 310.35 + 295.96 = 1099.83$

Output

786.69	743.57	768.35
784.03	947.02	767.75
968.56	1099.83	657.79

6) Data una rete neurale MLP a 3 livelli <u>con bias</u> composta da:

- 16 neuroni per l'input layer
- 10 neuroni per l'hidden layer
- 5 neuroni di output

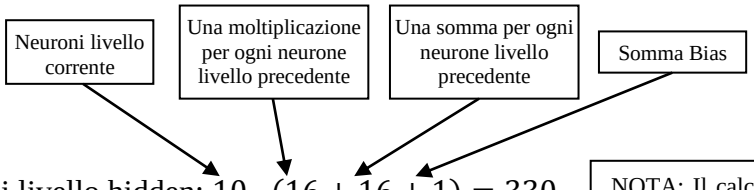
Quante somme e moltiplicazioni sono necessarie per il passo forward di un generico pattern trascurando le operazioni effettuate dalla funzione di attivazione? Motivare la risposta riportando anche il numero di operazioni per livello.

<u>Svolgimento</u>

Per ogni neurone del livello corrente si deve calcolare la seguente formula:

$$net_i = \sum_{j=1..d} w_{ji} \cdot in_j + w_{0i}$$

che comprende una moltiplicazione e una somma per ogni neurone del livello precedente più la somma finale del bias. Pertanto:



Numero operazioni livello hidden: $10 \cdot (16 + 16 + 1) = 330$

Numero operazioni livello di output: $5 \cdot (10 + 10 + 1) = 105$

<u>NOTA:</u> Il calcolo a lato è eseguito trascurando il fatto che un'implementazione ottimizzata della sommatoria potrebbe evitare una somma per il primo elemento della sommatoria (somma con 0).

Totale: **435 (225 somme e 210 moltiplicazioni)**

7) Supponendo di utilizzare *K-fold Cross-Validation* con $K = 5$ per suddividere 20000 pattern in *training* e *validation set*, quanti diversi addestramenti (*run*) vengono effettuati? Ad ogni *run* quanti pattern vengono utilizzati per il training e quanti per la validazione?

Svolgimento

Con $K = 5$ i 20000 pattern sono suddivisi in 5 partizioni da 4000 pattern l'una. Verranno eseguiti 5 addestramenti (*run*) utilizzando ogni volta una partizione diversa come *validation set* e le 4 restanti come *training set*. Pertanto ad ogni *run* verranno utilizzati 16000 pattern per il training e 4000 pattern per la validazione.