Machine Learning	**Matricola: ____________________**
19-Lug-2024 (90 minuti)	**Cognome: __________ Nome: ____________**

1) Nell'ambito delle reti neurali che cosa si intende per problema del vanishing gradient? Come può essere risolto?

Dispense "Deep Learning (parte 1)"

2) La posizione e forma dell'ellissoide di una distribuzione multinormale come è influenzata da μ e Σ?

Dispense "Classificazione (parte 1)"

3) Come può essere matematicamente definita la loss function (su un singolo pattern) per l'addestramento di una rete neurale?

Dispense "Reti neurali"

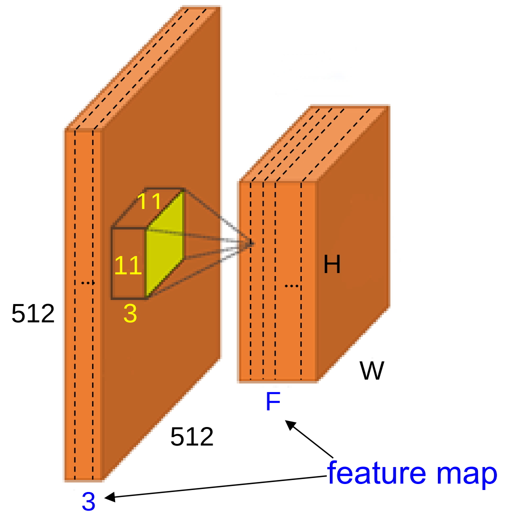
4) Con quali tecniche si può estendere SVM da 2 a più classi?

Dispense "Classificazione (parte 2)"

---

5) Dati un volume di input e un livello di convoluzione in una CNN, aventi le seguenti caratteristiche:

- *Volume Input*: $3 \times 512 \times 512$
- 128 filtri di dimensione $3 \times 11 \times 11$
- *Stride: 3*
- *Padding: 3*



Si calcoli motivando la risposta:

- La dimensione del volume di output: $F \times W \times H$;
- il numero totale di connessioni e di pesi del livello (<u>considerando i bias</u>).

<u>Svolgimento</u>

Il numero di feature map F del volume di output è pari al numero di filtri utilizzati: 128.
Essendo larghezza e altezza uguali tra loro sia nel volume di input che nei filtri, le dimensioni spaziali del volume di output (W e H) saranno anch'esse uguali tra loro.

$$W = H = \frac{D_{IN} - D_F + 2 \cdot Padding}{Stride} + 1 = \frac{512 - 11 + 2 \cdot 3}{3} + 1 = \frac{507}{3} + 1 = 170$$

Ogni neurone del livello di output ($128 \times 170 \times 170$) è connesso con tanti neuroni del livello di input pari alla dimensione del filtro ($3 \times 11 \times 11$).
Pertanto, **il numero totale di connessioni** è $(128 \times 170 \times 170) \cdot (3 \times 11 \times 11) = 1\,342\,809\,600$.

Il **numero totale di pesi**, invece, risulta molto più piccolo giacché in una CNN i pesi di ciascun filtro sono condivisi da tutti i neuroni contenuti in una stessa feature map. Visto che il numero di feature map è uguale a 128, e il numero di elementi di ciascun filtro è pari a $(3 \times 11 \times 11)$, il numero totale di pesi (considerando il bias) è $(3 \times 11 \times 11 + 1) \times 128 = 46\,592$.

6) Dato un insieme di pattern bi-dimensionali composto da 4 elementi:

$$\mathbf{p_1} = \begin{bmatrix} -1.6 \\ 4.4 \\ -4.0 \end{bmatrix}, \mathbf{p_2} = \begin{bmatrix} 4.3 \\ 2.1 \\ -0.2 \end{bmatrix}, \mathbf{p_3} = \begin{bmatrix} -3.3 \\ 0.6 \\ 3.9 \end{bmatrix}, \mathbf{p_4} = \begin{bmatrix} 3.2 \\ -1.6 \\ 1.8 \end{bmatrix}$$

Effettuare la prima iterazione dell'algoritmo K-means supponendo di dover raggruppare i pattern in 2 cluster rappresentati dai seguenti centroidi:

$$\mathbf{c_1} = \begin{bmatrix} -3.7 \\ 4.3 \\ -1.2 \end{bmatrix}, \mathbf{c_2} = \begin{bmatrix} 5.0 \\ -3.7 \\ 1.4 \end{bmatrix}$$

Riportare il cluster di appartenenza di ogni pattern e le coordinate dei nuovi centroidi calcolate in seguito all'iterazione svolta.

<u>Svolgimento</u>

Un'iterazione del K-means consiste i) nell'assegnare ogni pattern al cluster per cui è minima la distanza dal corrispondente centroide, e ii) nell'aggiornare i centroidi come media dei pattern assegnati al cluster corrispondente.
Utilizzando la distanza euclidea (si può utilizzare la distanza euclidea al quadrato evitando di calcolare la radice quadrata) i pattern vengono attribuiti ai cluster come segue:

$Cluster_1 = \{\mathbf{p_1}, \mathbf{p_3}\}$ $Cluster_2 = \{\mathbf{p_2}, \mathbf{p_4}\}$

Di conseguenza, i nuovi centroidi saranno:

$$\mathbf{c_1}' = \begin{bmatrix} -2.5 \\ 2.5 \\ -0.1 \end{bmatrix}, \mathbf{c_2}' = \begin{bmatrix} 3.8 \\ 0.3 \\ 0.8 \end{bmatrix}$$

7) Per il training di un classificatore binario SVM si procede con una *grid search* combinata a *k-fold cross-validation* (con $k = 15$). Nell'ottica di voler valutare le seguenti combinazioni di kernel/iperparametri:

1. Lineare
   - $C = \{1, 0.1, 0.01, 0.001\}$
2. RBF
   - $C = \{1, 0.1, 0.01, 0.001\}$
   - $\gamma = \{0.5, 0.05, 0.005\}$
3. Polinomiale
   - $C = \{1, 0.1\}$
   - $degree = \{2,3,5\}$
   - $\gamma = \{0.3, 0.2, 0.1\}$
   - $coef0 = \{0\}$

Si determini il numero complessivo di addestramenti da effettuare motivandone la risposta.

<u>Svolgimento</u>

Il numero di combinazioni di iperparametri per il kernel lineare è pari a 4, per il kernel RBF è uguale a $4 \cdot 3 = 12$ mentre per il kernel polinomiale è uguale a $2 \cdot 3 \cdot 3 \cdot 1 = 18$ per un totale di $4 + 12 + 18 = 34$ combinazioni. Per ognuna di queste combinazioni, la *grid search* esegue la *k-fold cross-validation* per un totale di $34 \cdot 15 = 510$ addestramenti.