

# Assignment 3

Belotti Federico - 808708  
In team con Scarpellini Gianluca - 807541

8 dicembre 2019

## 1 Obiettivi

Il task assegnato per questo assignment consiste nell'ottimizzazione degli iperparametri di una rete neurale, con l'obiettivo di migliorare le performance in termini di accuracy della 10 fold cross-validation.

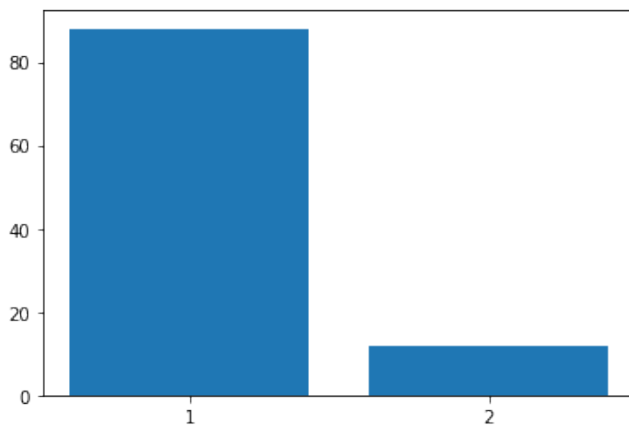


Figura 1: Distribuzione dataset

## 2 Dataset e preprocessing

Il dataset a disposizione è composto da 100 istanze, la cui distribuzione è rappresentata in figura 1.

Pur essendo molto sbilanciato non è stata applicata alcun tipo di data augmentation, né è stata applicata alcuna trasformazione.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	Class
0	-0.33	0.69	0	1	1	0	0.8	0	0.88	1
1	-0.33	0.94	1	0	1	0	0.8	1	0.31	2
2	-0.33	0.50	1	0	0	0	1.0	-1	0.50	1
3	-0.33	0.75	0	1	1	0	1.0	-1	0.38	1
4	-0.33	0.67	1	1	0	0	0.8	-1	0.50	2

Figura 2: Primi cinque record del dataset

### 3 Dettagli implementativi e risultati

L'obiettivo dell'assignment consiste nell'ottimizzare gli iperparametri di una rete neurale al fine di migliorare le performance in termini di accuracy della 10 fold cross-validation. Gli esperimenti si suddividono in due step differenti e gli iperparametri da ottimizzare durante i due step sono riassunti nella seguente tabella:

Step	Parametro	Intervallo
Step 1 e 2	Learning rate	$[0.01, 0.1] \in \mathbf{R}$
Step 1 e 2	Momentum	$[0.1, 0.9] \in \mathbf{R}$
Step 2	# neuroni per layer	$[1, 5] \in \mathbf{N}$

Tabella 1: Iperparametri da ottimizzare e rispettivo dominio<sup>1</sup>

La rete neurale è composta da 2 layer nascosti e gli iperparametri che non vengono ottimizzati durante il processo sono lasciati con i valori di default assegnati da `scikit-learn`.

#### 3.1 Step 1

Durante il primo step vengono ottimizzati il learning rate iniziale e il momentum; i risultati ottenuti vengono inoltre confrontati con le procedure di Random e Grid Search. In questo primo step si è scelto il *Gaussian Process* come modello surrogato, testato con due differenti funzioni d'acquisizione: *Upper Confidence Bound* e *Expected Improvement*. Ad entrambi i modelli surrogati sono stati forniti 5 esempi iniziali generati mediante Latin Hypercube Sampling e sono stati ottimizzati per 20 iterazioni totali, mentre le procedure di Random e Grid Search hanno testato rispettivamente 25 diverse configurazioni. I risultati ottenuti sono rappresentati nelle seguente tabella:

<sup>1</sup>Nel primo step il numero di neuroni è tenuto fisso a 4 per il primo layer e 2 per il secondo

Procedura	Lr	Momentum	Best seen
GP/EI	0.09	0.26	0.88
GP/UCB	0.09	0.26	0.88
Random search	0.05	0.49	0.88
Grid search	0.01	0.1	0.88

Tabella 2: Risultati step 1

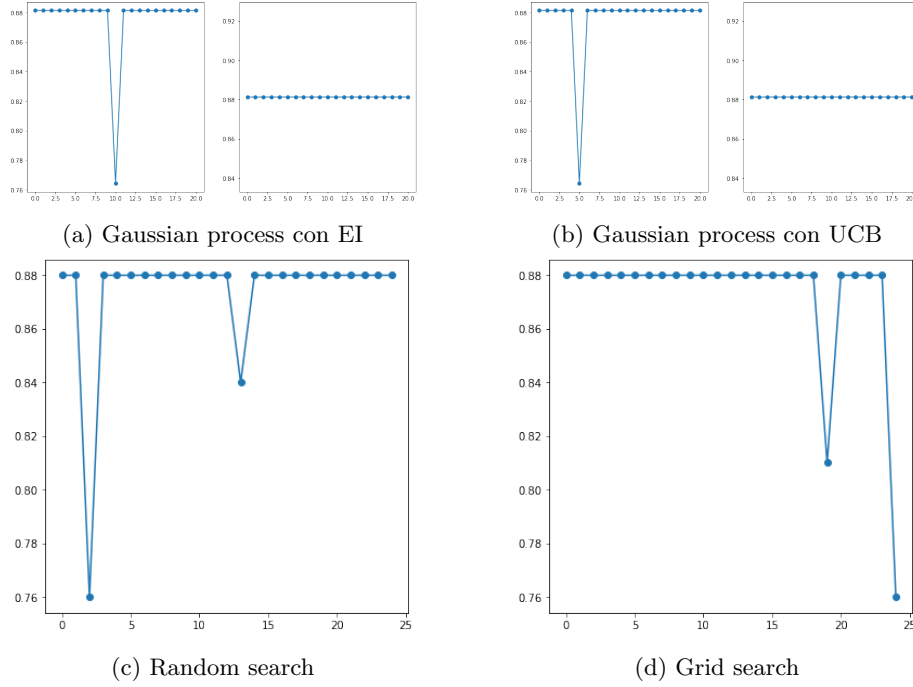


Figura 3: Risultati step 1

Nei grafici 3a e 3b (lo stesso varrà anche per i successivi grafici 5a e 5b), a sinistra è rappresentata l'accuracy media per i punti selezionati dalla funzione di acquisizione durante ogni iterazione, a destra si tiene traccia del miglior valore d'accuracy ad ogni iterazione.

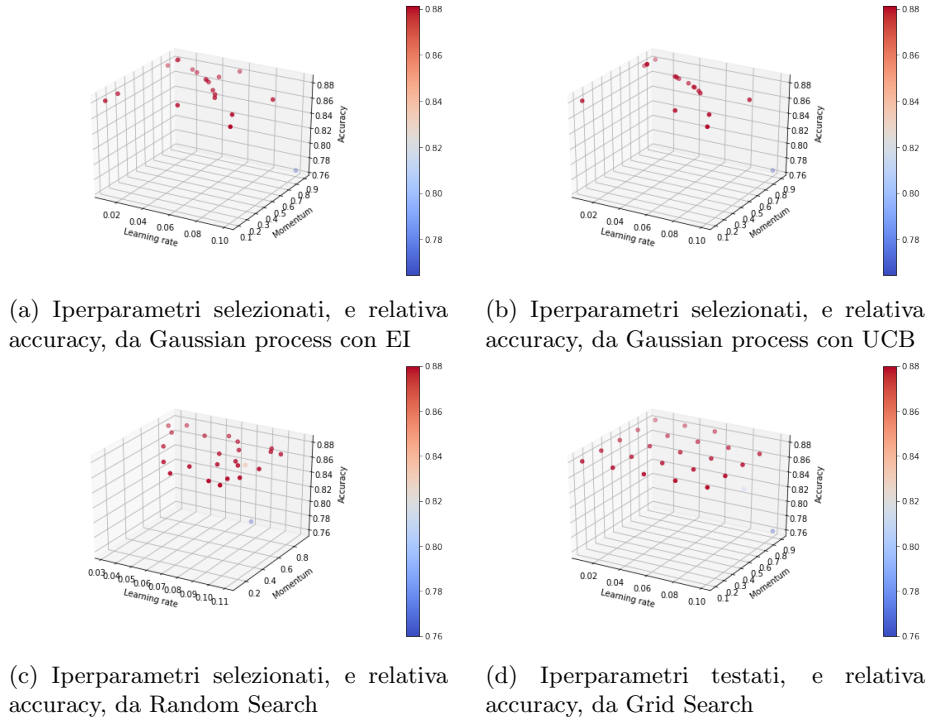


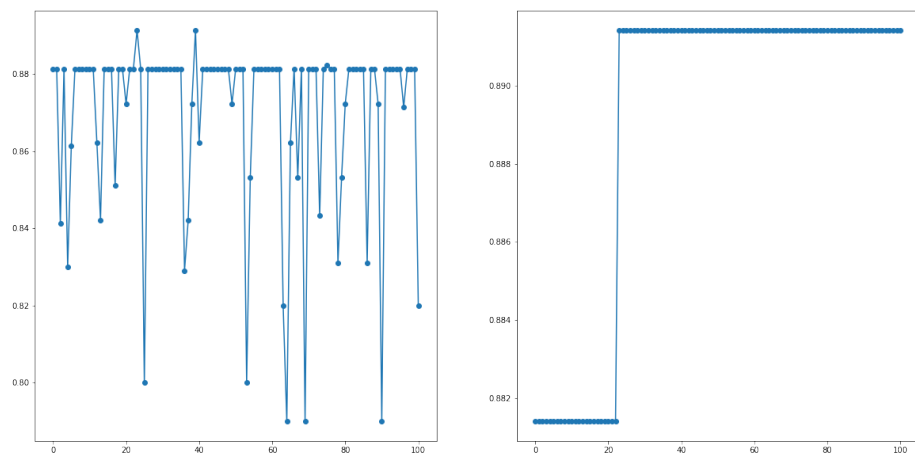
Figura 4: Risultati step 1

### 3.2 Step 2

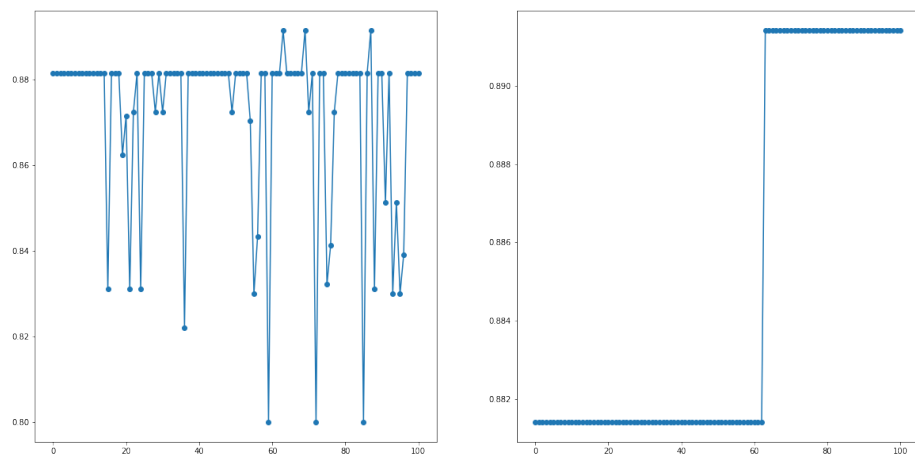
Durante il secondo step vengono ottimizzati, oltre al learning rate iniziale e il momentum, anche il numero di neuroni nei primi due layer, come specificato nella tabella 1. In questo secondo step si è scelto una *Random Forest* come modello surrogato, testato con due differenti funzioni d'acquisizione: *Upper Confidence Bound* e *Expected Improvement*. Ad entrambi i modelli surrogati sono state fornite 10 configurazioni iniziali generate mediante Latin Hypercube Sampling e sono stati ottimizzati per 100 iterazioni totali. I risultati ottenuti sono rappresentati nelle seguente tabella:

Procedura	Lr	Momentum	Neuroni layer 1 e 2	Best seen
RF/EI	0.08	0.23	4 e 3	0.89
RF/UCB	0.08	0.19	4 e 3	0.89

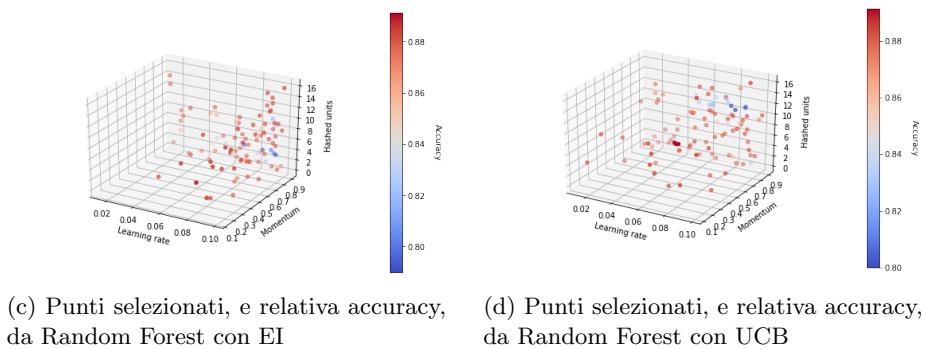
Tabella 3: Risultati step 2



(a) Random Forest con EI



(b) Random Forest con UCB



(c) Punti selezionati, e relativa accuracy, da Random Forest con EI

(d) Punti selezionati, e relativa accuracy, da Random Forest con UCB

Figura 5: Risultati step 2

Sull'asse z delle figure 5c e 5d vengono rappresentati gli interi univoci assegnati ai punti bidimensionali della forma ( $\#$  neuroni 1°layer,  $\#$  neuroni 2°layer). I mapping univoci sono rappresentati nella seguente tabella:

Coppia	Intero	Coppia	Intero
(4, 4)	1	(4, 1)	9
(3, 2)	2	(4, 2)	10
(1, 2)	3	(2, 2)	11
(4, 3)	4	(2, 3)	12
(3, 3)	5	(1, 4)	13
(1, 1)	6	(3, 1)	14
(1, 3)	7	(2, 4)	15
(3, 4)	8	(2, 1)	16

Tabella 4: Mapping univoci

## 4 Analisi dei risultati

Dai risultati della tabella 2 si nota come per il primo step entrambi i processi gaussiani, anche se con differenti funzione d'acquisizione, convergano alle stesse soluzioni ottime, e dalle figure 4a e 4b si vede come lo spazio di ricerca degli iperparametri sia molto simile; Random e Grid Search forniscono invece differenti valori degli iperparametri con la stessa accuracy ottima.

Dai risultati in tabella 3 si nota come i valori degli iperparametri "Lr" e "Momentum" siano tra loro, ed anche rispetto ai valori della tabella 2, simili; l'accuracy è invece migliorata di un punto percentuale ottenendo l'89%: avendo ottimizzato nel secondo step anche il numero di neuroni per layer e date le considerazioni di cui prima, si può dedurre che in questo esperimento, a "parità" di learning rate e momentum, il numero di neuroni per layer influenza positivamente l'accuracy media del modello; inoltre si può notare come, con il numero di neuroni per layer "ottimi", l'architettura globale della rete assuma la configurazione standard (ad imbuto) vista in letteratura: 9 (Input)  $\rightarrow$  4  $\rightarrow$  3  $\rightarrow$  1; infine i due processi differiscono sostanzialmente nel numero d'iterazione in cui trovano gli iperparametri ottimi: intorno alla 20-esima per Random Forest con EI ed intorno alla 70-esima per Random Forest con UCB.