

# Week 5 — Assignment Submission

Gianluca Scarpellini - 807541 - g.scarpellini1[at]disco.unimib.it  
(In team con Federico Belotti - 808708)

8 dicembre 2019

## Indice

<b>1 Task</b>	<b>1</b>
<b>2 Dataset e preprocessing</b>	<b>2</b>
<b>3 Modelli e frameworks per HPO</b>	<b>2</b>
3.1 Pygpgo . . . . .	2
3.2 Step 1 . . . . .	3
3.3 Step 2 . . . . .	3
<b>4 Esperimenti e risultati</b>	<b>3</b>
4.1 Step 1 . . . . .	3
4.2 Step 2 . . . . .	4
<b>5 Conclusione</b>	<b>5</b>

## 1 Task

Il task dell'assignment consiste nell'effettuare *ricerca di iperparametri* del processo di training di una rete neurale. Il dataset impiegato è *fertility*, la cui composizione viene meglio approfondita in sezione 2. Il task si divide in due step e per entrambi prevede l'impiego di *processi gaussiani* per effettuare Hyperparameters Optimazation (HPO). L'approccio a processi gaussiani viene confrontato, nello step 1, con un approccio deterministico (grid search) e un approccio random (random search). Le specifiche relative agli esperimenti effettuati sono riportati in sezione 3, mentre i risultati commentati e i relativi grafici sono riportati e commentati in sezione 4.2. In sezione 5 traggio le mie conclusioni circa l'approccio seguito e indago i possibili sviluppi futuri.

## 2 Dataset e preprocessing

Il dataset impiegato nel present assignment è *fertility*, il quale è liberamente disponibile sulla piattaforma OpenML. Si tratta di un dataset di 100 istanze e 10 attributi (di cui 9 feature e un target binario) relative a condizioni mediche relative alla fertilità. Gli attributi del dataset si riferiscono ai fattori ambientali dell'acquisizione dei dati (come la stagione) e a eventi tramutaci o cattive abitudini che potrebbero aver inciso sul tasso di fertilità del paziente. In figura 1 riporto le prime istanze estratte dal dataset e la distribuzione delle classi. Si noti difatti che il dataset è ridotto e sbilanciato verso una diagnosi **normale**: tali condizioni rendono difficoltoso un approccio di ML.

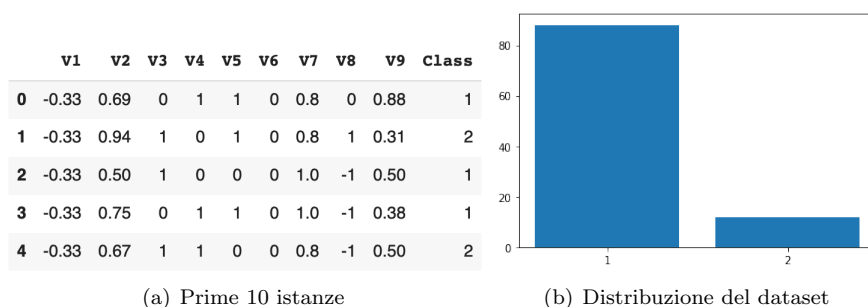


Figura 1: Dataset fertility

## 3 Modelli e frameworks per HPO

L'ottimizzazione di iperparametri fa uso di modelli non parametrici derivati dalla distribuzione gaussiana: si tratta dei *processi gaussiani*. Si definisce *processo gaussiano* un processo stocastico rappresentato da una *funzione media* e un set di funzioni che rappresentino la varianza rispetto alla media. Si può vedere tale definizione come una generalizzazione della distribuzione gaussiana multivariata a un numero infinito di variabili aleatorie tali che, preso un subset finito delle stesse, la loro distribuzione congiunta risulti essere una normale multivariata.

### 3.1 Pygpgo

Per portare correttamente a compimento i due step del task assegnatoci il mio team ha ridefinito diverse funzioni della nota libreria *pygpgo*. In particolare, abbiamo introdotto la possibilità di assegnare a priori i punti da cui partire per effettuare HPO e introdotto, nella funzione di *update*, alcuni step di data visualization per ottenere i grafici in sezione 4.2.

### 3.2 Step 1

Il primo step dell'assignment prevede di individuare i valori di *learning rate* e di *momentum* partendo da 5 configurazioni generate a priori con l'algoritmo LHS. La rete neurale impiegata è invece stata fissata a priori: si tratta di una feed forward network con due layer intermedi rispettivamente di 4 e 2 neuroni. Per rispettare la consegna abbiamo impiegato *SGD* quale algoritmo di ottimizzazione. Nello step 1 abbiamo deciso di impiegare i *gaussian process* quale modello surrogato. Abbiamo confrontato due diverse funzioni di acquisizione (UCB ed Expected Improvement) per il modello GP con l'approccio Grid Search e Random Search. I risultati sono riportati in 4.1.

### 3.3 Step 2

Il secondo step prevede di impiegare 10 configurazioni iniziali e di introdurre il numero di neuroni dei due layer (in un range compreso tra 1 e 5) quali parametri da ottimizzare. In questo step abbiamo impiegato un modello *Random forest* e abbiamo confrontato due funzioni di acquisizione (UCB ed Expected Improvement). I risultati sono riportati in sezione 4.2.

## 4 Esperimenti e risultati

### 4.1 Step 1

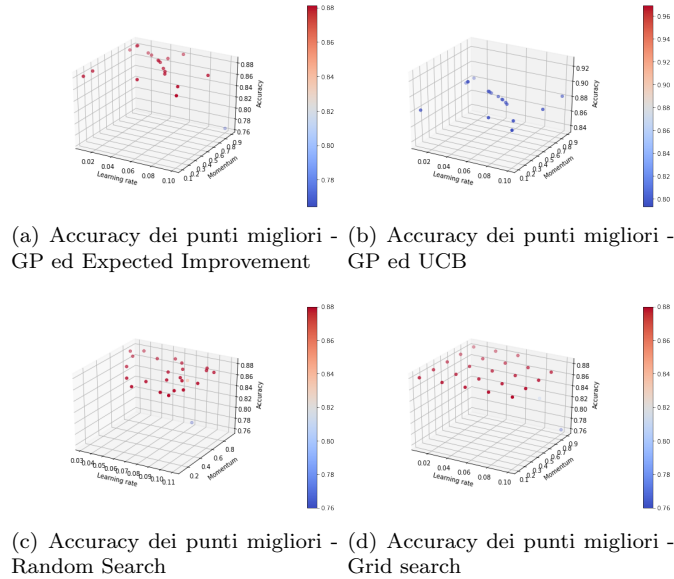


Figura 2: Grafici relativi alle accuracy dei 4 approcci confrontati

In figura 3 riporto i grafici di accuracy relative ai 4 approcci utilizzati. Entrambi gli approcci gaussiani hanno indagato tra 20 possibili configurazioni, mentre gli approcci random e grid search sono stati eseguiti per 25 iterazioni. I risultati riportati mostrano pcome entrambi gli approcci gaussiani individuano il set di iperparametri migliori sin dalle prime iterazioni (in figura 3(a) e 3(b)) e apprendono a mantenersi nella configurazione migliore con un accuracy di 90%. Gli approcci grid search e random search non hanno un modello della storia appresa, e oscillano pertanto tra configurazioni più o meno ottimali. In figura 2 sono rappresentati i punti migliori individuati dai 4 approcci con i relativi valori di accuracy.

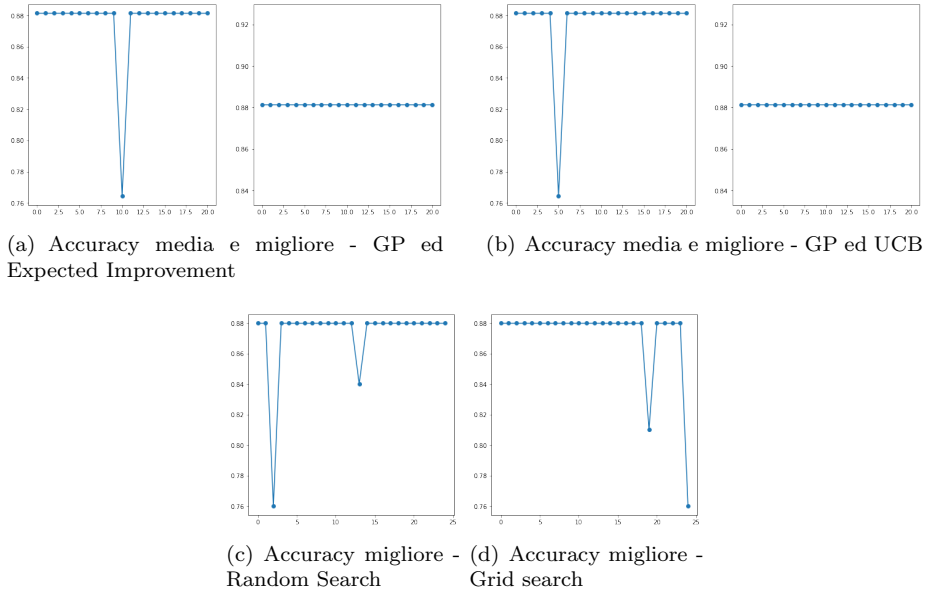


Figura 3: Grafici relativi alle accuracy dei 4 approcci confrontati

## 4.2 Step 2

In figura 5 riporto i grafici di accuracy degli iperparametri per gli approcci gaussiani con funzioni di acquisizione Expected Improvement e UCB su 100 iterazioni. In particolare si noti una maggiore ottimalità per l'approccio di Expected Improvement, che individua una configurazione ottimale con un minor numero di iterazioni rispetto all'approccio UCB. In figura 4 sono inoltre riportati le rappresentazioni di accuracy relative ai punti migliori. Per rappresentare nello spazio 3D i 4 iperparametri da ottimizzare e la relativa accuracy faccio uso di un *hashing* della coppia di valori discreti relativi al numero di neuroni; l'accuracy è qui rappresentata tramite la tinta di colore.

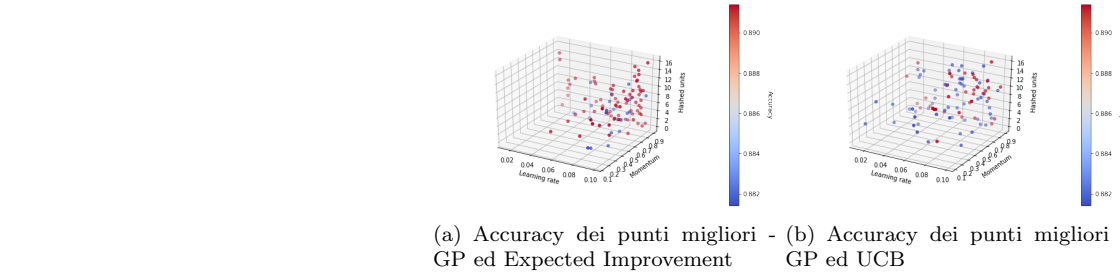


Figura 4: Grafici relativi alle accuracy dei 2 approcci confrontati

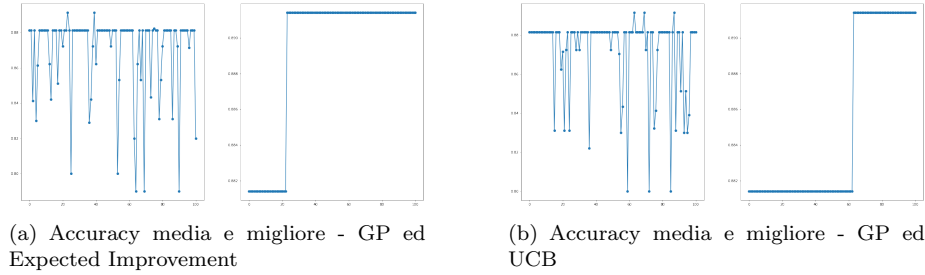


Figura 5: Grafici relativi alle accuracy dei 2 approcci confrontati

## 5 Conclusione

In conclusione, la ricerca ottimale di iperparametri è oggi utile al fine di individuare, senza l'ausilio di esperti di dominio o di ML, i parametri utili a raggiungere l'ottimo per un dato task. In particolare, l'impiego di approccio probabilistici, in confronto ai tradizionali grid search e random search, permettono di meglio impiegare le conoscenze di stato acquisite nella ricerca dei parametri. Nel presente lavoro abbiamo confrontato ed esplorato diversi modelli di HPO e diverse funzioni di acquisizione. La possibilità già divenuta presente di esplorare la struttura stessa di un modello di DL tramite NAS contribuisce al processo di *democratizzazione* del DL.