

Report dell'Assignment 5 di Advanced Machine Learning

Paolo Mariani 800307

Dicembre 2019

1 Task

L'obiettivo dell'Assignment 5 è effettuare l'ottimizzazione degli iperparametri della rete neurale che svolge un compito di classificazione binaria attraverso due task:

- Il **primo compito** prevede di ottimizzare l'iperparametro *Learning Rate* (che possiede un dominio di valori pari a $[0.01, 0.1]$) e l'iperparametro *Momentum* (che possiede un dominio di valori pari a $[0.1, 0.9]$), vincolando la struttura della rete neurale (Multilayer Perceptron) a 2 strati nascosti di 4 e 2 neuroni rispettivamente;

L'ottimizzazione deve essere effettuata attraverso un processo di **Sequential Model Based Optimization (SMBO)** che prevede l'utilizzo di un modello surrogato (*Gaussian Process* o *Random Forest*) che copra 25 diverse configurazioni di parametri (di cui 5 iniziali casuali impostate in funzione di un seed pari a 1996, mentre le altre 20 stimate dal modello autonomamente).

Il procedimento deve essere ripetuto due volte con due funzioni di acquisizione diverse ("**Expected Improvement**" e "**Probability Improvement**") e poi entrambi devono essere confrontati con due ulteriori processi di ottimizzazione ottenuti con **Grid Search** e **Random Search**.

- Il **secondo compito** prevede che si ottimizzino 4 diversi iperparametri:
 - *Learning Rate* (con dominio $[0.01, 0.1]$)
 - *Momentum* (con dominio $[0.1, 0.9]$)
 - *Numero di neuroni del primo strato nascosto della rete* (da 1 a 5)

– *Numero di neuroni del secondo strato nascosto della rete* (da 1 a 5)

Si vincola l'ottimizzazione attraverso la Sequential Model Based Optimization con modello surrogato "Random Forest", 10 configurazioni iniziali (impostate con seed 1996) e che si ripeta due volte per ciascuna delle due funzioni di attivazione: "Expected Improvement" e "Probability Improvement" per poi procedere ad effettuarne un confronto.

Per entrambi i compiti si prevede l'utilizzo della tecnica della Cross Validation con 10 fold.

1.1 Dataset

Il Dataset utilizzato per questo Assignment è "fertility" proveniente dalla collezione UCI (fertility dataset), possiede 100 osservazioni relative ad un esperimento legato alla fertilità di cui 88 appartengono alla classe 1 e i restanti 12 alla classe 2. I relativi attributi sono i seguenti:

- **V1**, Stagione in cui è stata effettuata l'analisi:
 - winter
 - spring
 - Summer
 - fall
- **V2**, Età al momento dell'analisi: valori da 18 a 36
- **V3**, Malattie infantili (ad esempio, varicella, morbillo, parotite, polio): variabile binaria (2 = no, 1 = sì)
- **V4**, Incidenti o traumi gravi: (2 = no, 1 = sì)
- **V5**, Interventi chirurgici: (2 = no, 1 = sì)
- **V6**, Febbre alta nell'ultimo anno:

- meno di tre mesi fa
- più di tre mesi fa
- no
- **V7**, Consumo di alcool:
 - più volte al giorno
 - ogni giorno
 - più volte a settimana
 - una volta a settimana
 - quasi mai o mai
- **V8**, Ogni quanto il paziente fuma:
 - mai
 - occasionalmente
 - quotidianamente
- **V9**, numero di ore passate in posizione seduta
- **Class**, variabile dipendente con valori:
 - Esito normale (1)
 - Esito alterato (2)

Il Dataset è stato inserito in una struttura DataFrame Pandas per garantire una gestione ed analisi facile dei dati.

2 Preprocessing

È stata effettuata una brevissima attività di pre-processing con il fine di trasformare le colonne del dataset in vettori numpy, separando la variabile target "Class" dal resto delle variabili esplicative (V1 fino a V9).

3 Primo Compito

Il primo compito è stato svolto partendo dalla definizione di una funzione obiettivo da ottimizzare: in particolare al suo interno sono scritte istruzioni per poter definire una rete neurale, passare gli iperparametri al suo interno ed effettuare la cross validation. La funzione restituisce il valore medio della accuracy registrata per ogni iterazione della cross validation. Il **primo processo** di ottimizzazione svolto è quello dotato di 5 configurazioni iniziali e funzione di acquisizione "Expected Improvement", che ha permes-

so di raggiungere il 90% di Accuracy con i seguenti valori degli iperparametri:

- Learning Rate: 0.023568486291425158
- Momentum: 0.5441283633693811

Il **secondo processo** di ottimizzazione ha sfruttato le stesse 5 configurazioni iniziali del processo precedente ma ha utilizzato una funzione di acquisizione "Probability Improvement" che ha permesso di raggiungere un valore di Accuracy pari a 89%. I parametri ottimali che ha identificato sono:

- Learning Rate: 0.011431165290502747
- Momentum: 0.5928497535262441

Mettendo i due precedenti processi a confronto si evidenzia una Accuracy superiore per il secondo processo, che si distingue dal primo solamente a partire dalla quindicesima iterazione. Il primo processo non migliora il valore ottenuto dalla seconda configurazione iniziale casuale.

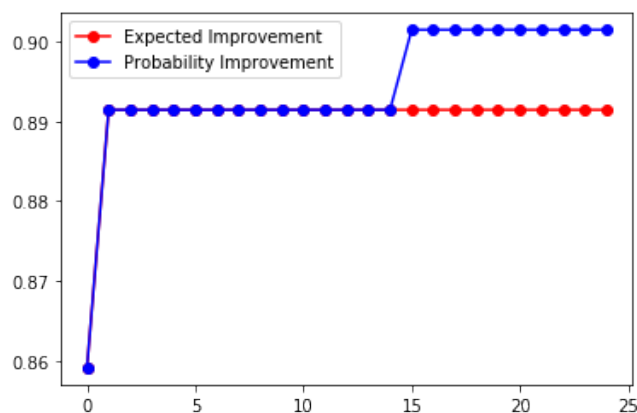


Figura 1: confronto grafico tra i due processi

In aggiunta rispetto ai due processi sono stati svolti due processi separati. Un processo di ottimizzazione è stato effettuato tramite **Grid Search**, impostando come parametri 5 valori di Learning Rate e 5 di Momentum (estratti tramite la funzione linspace di Numpy) che hanno generato in totale 25 combinazioni diverse: è stata ottenuta una Accuracy pari a 87,5% con iperparametri:

- Learning Rate: 0.01
- Momentum: 0.1

L'ultimo processo di ottimizzazione è stato implementato tramite **Random Search**, impostando uno spazio di 1000 valori assumibili per ogni iper-

parametro; A seguito di 25 iterazioni il processo ha ottenuto una Accuracy massima identica a quella del precedente processo (0.875), con i seguenti iperparametri:

- Learning Rate: 0.0290990990990991
- Momentum: 0.8751751751751752

Una valutazione complessiva dei risultati confrontando i modelli viene suggerita dalla seguente immagine:

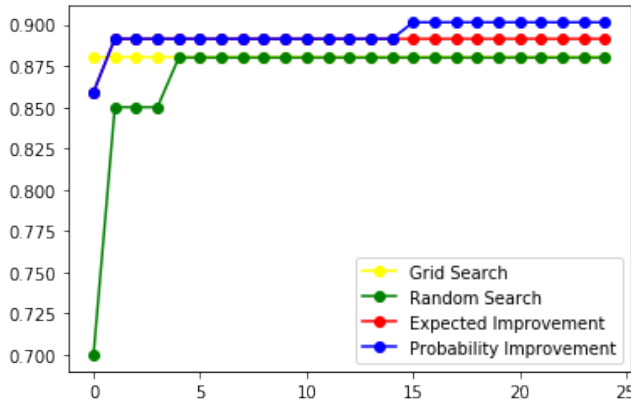


Figura 2: confronto grafico tra i quattro processi

Si nota che i processi effettuati tramite modello surrogato Gaussian Process con funzioni "Expected Improvement" e "Probability Improvement" possiedono un andamento identico per le prime 15, poi il secondo migliora trovando la configurazione ottimale migliore tra i 4 processi. Il processo Grid Search e Random Search non raggiungono valori superiori al 87,5% non permettendo al modello di migliorarsi; il valore di Accuracy è estremamente vicino alla percentuale di osservazioni appartenenti alla classe 1: non avendo a disposizione ulteriori informazioni sui risultati non si può convalidare l'ipotesi di Zero Rule.

4 Secondo Compito

Nel secondo compito si procede ad effettuare una ottimizzazione di 4 parametri, e per questo motivo la funzione obiettivo creata per il task precedente è stata opportunamente modificata per permettere l'inserimento dei parametri aggiuntivi legati al numero di neuroni della rete neurale.

I due processi di ottimizzazione differiscono per la funzione di acquisizione, che risultano essere ancora

"Expected Improvement" e "Probability Improvement", e possiedono entrambi le stesse 10 configurazioni iniziali. Ciascuno di essi ha svolto 100 iterazioni in totale, determinando i valori ottimali degli iperparametri.

Il primo processo ha permesso di raggiungere un valore di Accuracy massima pari a 89%, definendo come ottimali i seguenti iperparametri:

- Learning Rate: 0.02261380244477139
- Momentum: 0.6920724059905914
- Numero di neuroni del primo strato nascosto della rete: 1
- Numero di neuroni del secondo strato nascosto della rete: 2

Il secondo processo invece ha permesso di ottenere una Accuracy pari a 88%, stabilendo come iperparametri ottimali:

- Learning Rate: 0.07883391130986059
- Momentum: 0.11036849951367192
- Numero di neuroni del primo strato nascosto della rete: 1
- Numero di neuroni del secondo strato nascosto della rete: 1

Si può effettuare un confronto tra i due modelli dal punto di vista grafico attraverso la seguente immagine:

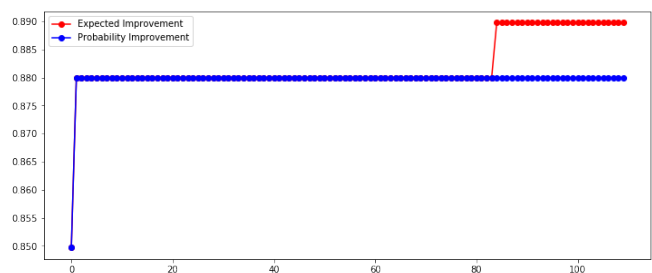


Figura 3: confronto grafico tra i due processi su 100 iterazioni

Dall'immagine si osserva che a partire dalla ottantacinquesima iterazione si ottiene il valore di configurazione migliore dato dal processo "Expected Improvement". Per gran parte del processo di ottimizzazione sequenziale non avviene alcun miglioramento rispetto alle 10 configurazioni iniziali, per poi migliorare una sola volta per trovare il valore ottimo.

5 Conclusioni

L'ottimizzazione effettuata in entrambi i compiti ha sollevato dei dubbi relativi alla validità dei risultati ottenuti, in quanto tutti i valori risultano molto vicini tra loro e soprattutto molto vicini alla percentuale di osservazioni della classe 1 del dataset sul totale di osservazioni: si ritiene infatti che il Dataset sia estremamente ridotto per poter effettuare una ottimizzazione sufficientemente valida e che a

questo scopo dovrebbe essere effettuata una manipolazione del dataset col fine di bilanciare le due classi (Oversampling della classe inferiore 2), o in alternativa procedere con l'ottenimento di osservazioni aggiuntive.

Non essendo elencata all'interno della specifica non è stata effettuata alcuna operazione di preprocessing ulteriore a quella già effettuata, e di conseguenza è stato deciso di non effettuare oversampling.