



# UNIVERSITÀ DI PARMA

Dipartimento di Ingegneria e Architettura

Corso di Laurea Triennale in Ingegneria Informatica, Elettronica e delle  
Telecomunicazioni

---

## CLASSIFICAZIONE DI SEGNALI MEDIANTE PROGRAMMAZIONE GENETICA MODULARE

Relatore:

Prof. Stefano Cagnoni

Tesi di Laurea di:

Arianna Cella

Correlatori:

Dott.ssa Giulia Magnani

ANNO ACCADEMICO 2022/2023

Questo lavoro di tesi descrive lo studio e lo sviluppo di un algoritmo basato sulla programmazione genetica modulare per la classificazione dei segnali.

Il progetto ha due obiettivi. Il primo è utilizzare la programmazione genetica (GP) per effettuare una preelaborazione dei dataset al fine di migliorare le prestazioni di un classificatore random forest. Il secondo obiettivo è realizzare un approccio modulare basato sulla GP per realizzare questa pre-elaborazione. In particolare, si intende implementare una versione di GP modulare che possa semplificare l'implementazione su FPGA del sistema, rendendo più compatte sia la ricerca che la descrizione degli alberi del sistema grazie all'utilizzo di moduli che implementano funzioni di alto livello.

I segnali sono sequenze temporali di dati provenienti da diverse fonti e la loro classificazione, che consiste nell'identificare e assegnare una categoria o una classe a ciascun segnale, viene effettuata in base alle loro caratteristiche o a pattern distintivi.

La GP rientra nel campo degli algoritmi evolutivi ed è una tecnica di ottimizzazione ispirata ai principi dell'evoluzione biologica. Consiste nel far evolvere una popolazione iniziale di individui, ognuno rappresentante una funzione che risolve un problema dato, ottenendo generazione dopo generazione soluzioni sempre migliori attraverso la selezione dei migliori individui e l'applicazione degli operatori di crossover e mutazione. La selezione si basa sulla valutazione del grado di adattamento di ciascun individuo tramite una funzione di fitness che misura il livello di soddisfazione rispetto agli obiettivi desiderati.

In questo progetto, per implementare l'algoritmo evolutivo è stata usata la libreria DEAP (Distributed Evolutionary Algorithms in Python). La libreria offre una vasta gamma di funzioni e moduli utili per la realizzazione degli algoritmi evolutivi.

Fondamentale in un algoritmo evolutivo è la definizione della funzione di fitness, che ha il compito di valutare e assegnare un punteggio a ciascun individuo della popolazione. In questo caso specifico, l'obiettivo è trovare una trasformazione del segnale che permetta di ottimizzare le prestazioni di un classificatore Random Forest applicato ai dati trasformati. Ogni individuo viene utilizzato per trasformare il training set che addestra il modello di classificazione Random Forest, e il validation set, che valuta il modello addestrato.

Per valutare le prestazioni di questo approccio, il training set e validation set sono stati

generati a partire dal Dataset Digit, utilizzato per il riconoscimento di cifre scritte a mano, e da un dataset contenente segnali forniti da Physikalisch-Technische Bundesanstalt, utilizzato per la classificazione dei segnali normali o anormali dell'elettrocardiogramma. Il modello di classificazione usato è un classificatore Random Forest e la metrica di valutazione è la F1-score, che rappresenta la fitness dell'individuo e deve essere massimizzata nel corso delle iterazioni.

L'aspetto chiave di questo progetto è l'approccio modulare che viene implementato analizzando gli individui della popolazione e selezionando i sottomoduli di profondità 1 e 2 più frequenti. Tra questi sottomoduli vengono selezionati quelli con una fitness più elevata, che vengono poi inseriti nel set delle primitive utilizzate per generare gli individui delle popolazioni successive.

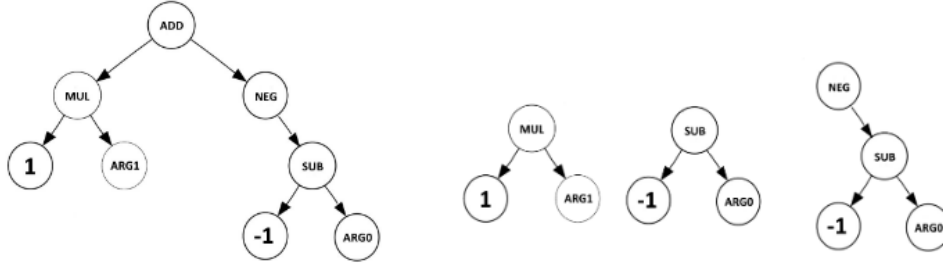


Figura 1: Sulla sinistra viene riportato un esempio di individuo, nello specifico  $add(mul(1, ARG1), neg(sub(-1, ARG0)))$ , mentre sulla destra vengono riportati i relativi sottomoduli che verranno estratti: quelli di profondità 1 saranno  $mul(1, ARG1)$  e  $sub(-1, ARG0)$ , mentre di profondità 2 avremo solo  $neg(sub(-1, ARG0))$

Per agevolare l'utente nell'esecuzione dell'algoritmo è stata sviluppata un'interfaccia grafica che consente di impostare i parametri principali dell'algoritmo evolutivo e il dataset su cui l'utente desidera lavorare. Una volta completata l'esecuzione, i risultati saranno disponibili in un file di testo specifico. Questo consente all'utente di esaminare e valutare i risultati in modo semplice e ordinato.

Attraverso l'interfaccia sviluppata, sono state eseguite diverse prove per testare varie combinazioni di parametri al fine di valutare quali di esse conducessero a risultati migliori. Dall'analisi dei risultati è emerso che aumentare il numero di individui in una popolazione e il numero di generazioni spesso porta a miglioramenti. Tuttavia, la variazione più significativa dei risultati si ha con la variazione del valore della dimensione del kernel e i risultati di F1 più elevati si osservano con dimensioni del kernel relativamente piccole.

Dai risultati ottenuti emerge che i filtri ottenuti attraverso la GP consentono di migliorare i risultati rispetto all'applicazione diretta del classificatore Random Forest ai segnali originali, nonostante quest'ultimo fornisca comunque buoni risultati più rapidamente. La GP modulare inoltre produce spesso miglioramenti rispetto alla GP standard, anche se non è stata sviluppata con tale obiettivo.

Come sviluppi futuri, si potrebbe esplorare ulteriormente l'ottimizzazione dell'approccio modulare sviluppando nuove strategie di selezione dei moduli o approcci più elaborati che combinino efficacemente i vantaggi della programmazione genetica con altri metodi di classificazione.