

Analisi comportamento coordinato tramite Neural Network

Leonardo Bigelli¹

¹l.bigelli2@campus.uniurb.it

10 maggio 2025

Riassunto

Questo progetto descrive l'implementazione di un approccio di Deep Learning, in particolare tramite l'utilizzo di una rete neurale di tipo denso, per l'addestramento e la classificazione del comportamento di due contenuti di account presenti su un social network generico.

1 Introduzione e obiettivi

Il Deep Learning sta avendo un notevole successo in questi ultimi anni. Uno dei suoi più grandi successi si basano sulla capacità di eseguire con successo classificazioni multi-classi o binarie. Nell'ambito di apprendere se contenuti di due account possono essere coordinati o meno, le reti neurali possono fornire un grande vantaggio. Questo per via della natura del problema, in quanto si può astrarre quest'ultimo come una classificazione binaria: coordinato o non-coordinato. Lo scopo del progetto è realizzare una rete neurale, più o meno semplice, con lo scopo di poter gestire questa casistica.

I modelli di Deep Learning necessitano di un grande quantitativo di dati per poter essere addestrati. Questo fattore è una criticità per via del dominio applicativo in questione, in quanto avere a disposizione dei dataset inerente al comportamento coordinato degli utenti nei social network comporta numerose problematiche relative alla privacy di quest'ultimi.

una particolare funzione specifica del tool **CooRT-weet**[Righetti et al., 2023] per generare dei datasets simulati con determinate caratteristiche.

Segue il codice per la creazione di un set di dati simulato con la funzione impiegata:

```
ris <- simulate_data(  
  approx_size = 200,  
  n_accounts_coord = 100,  
  n_accounts_noncoord = 100,  
  n_objects = 5,  
  min_participation = 3,  
  time_window = N,  
  lambda_coord = NULL,  
  lambda_noncoord = NULL  
)
```

dove, N è una lista di valore interi, precisamente composta da 10, 20, 30, 40, 50, 60, 70. Questi valori sono stati scelti per ottenere un dataset di training il più possibile generico. Per ciascun valore di `time_window` sono stati generati 10 set di dati per poi essere concatenati.

2 Setup

In questa sezione verranno delucidate le tecniche implementate per poter implementare il progetto in questione.

2.1 Dataset

Come espresso in precedenza, non sono disponibili molti dataset per questo ambito, e i pochi presenti nella letteratura non sono idonei per il tipo di applicazione per questo progetto.

Per ovviare a questo problema, si è scelto di utilizzare

2.2 Framework utilizzato

Per lo sviluppo della rete neurale, è stato impiegato il framework TensorFlow sviluppato da Google. Mentre per l'addestramento della rete stessa, è stata sfruttata l'applicazione Google Colab.

2.3 Rete Neurale

Come modello utilizzato per il progetto è stato scelto di usare una rete neurale molto semplice. Questa scelta è dovuta alla natura del problema, in quanto è

sufficiente una rete neurale densa totalmente connessa come utilizzato frequentemente come stato dell'arte. Tra i vari layers densi è stato aggiunto un layer di Dropout per evitare l'incorrenza di Overfitting, ovvero che la rete apprenda troppo perfettamente i dati di training portando ad una performance non ottima nel set di dati inerenti al testing. Considerando anche i dati sono simulati, l'aggiunta di layer di Dropout è fondamentale.

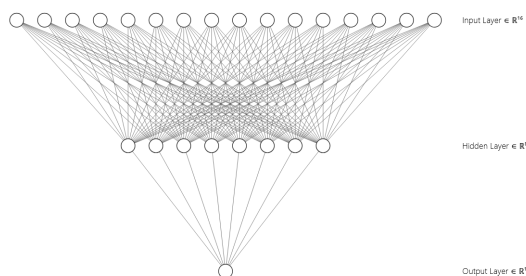


Figura 1: Modello della rete neurale.

Più nel dettaglio, la rete è composta da tre layers densi, quali i primi due con funzione di attivazione 'ReLU', mentre l'ultimo layer sfrutta la 'Sigmoid' per andare ad effettuare la classificazione.

Il modello è stato compilato con l'ottimizzatore 'Adam' e 'Binary Crossentropy' come funzione di perdita.

3 Risultati sperimentali

In questa sezione verranno spiegati i risultati ottenuti con relativi esperimenti.

La rete neurale è stata addestrata con 20 epoche, per poi essere successivamente valutato su una porzione di dati. Quest'ultimi non erano presenti nella porzione di dati utilizzati per l'addestramento del modello stesso.

Dalla valutazione effettuata, si evince un'accuratezza del 99,65% nel riconoscere se due contenuti sono coordinati o meno, per quanto riguarda una *Time Window* che varia tra i 10 e 70 secondi, a step di 10s. Il modello addestrato è stato valutato su un altro set di dati, totalmente scorrelato dall'insieme su cui è stato addestrato. Lo scopo di questo esperimento è comprendere il comportamento del modello anche su dati che non ne hanno caratterizzato il comportamento. Questo genere di approccio prende il nome di *Zero-Shot testing*, come stato dell'arte.

In particolare, è stato simulato un altro set di dati con *Time Window* compresa tra 80s e 100s, a step di 10s. Il modello ha raggiunto un'accuratezza leg-

germente più elevata, 99,66%, rispetto alla classica valutazione di quest'ultimo.

L'esperimento effettuato implica che il modello si presta egregiamente per inferire se due contenuti, di account diversi o meno, hanno un comportamento coordinato.

4 Disponibilità dati

Visto la scarsità di dati in questo ambito, tutti i dati utilizzati per addestrare e valutare la bontà del modello di Deep Learning, sono disponibili nella repository pubblica della piattaforma GitHub tramite il seguente indirizzo:

https://github.com/LeonardoBigelli/SNA_coordinated-behavior

5 Conclusioni

Questo progetto ha lo scopo di dimostrare che l'utilizzo delle tecniche di Deep Learning possono aiutare per comprendere se due contenuti di account, uguali o diversi, possono avere un comportamento coordinato o meno.

La carenza di dati pubblici in questo ambito è un problema rilevante. Per ovviare a ciò, si è utilizzata una funzione per ottenere un insieme di dati simulati con determinate caratteristiche.

Tramite lo sviluppo di una rete neurale è stato possibile dimostrare quanto descritto sopra, ottenendo un'accuratezza superiore al 99%.

5.1 Possibili sviluppi futuri e limitazioni

In futuro, se fossero presenti datasets pubblici riguardo al comportamento coordinato degli utenti sui Networks, sarebbe possibile eseguire dei test più specifici. In quanto tutti i risultati ottenuti e illustrati, sono ricavati su datasets simulati e quindi sintetici.

Riferimenti bibliografici

- [1] Righetti N., Balluff P. (2023). CoorRTweet: Coordinated Networks Detection on Social Media. R package version 2.1.0.
<https://CRAN.R-project.org/package=CoorRTweet>.