Esame di Sistemi Distribuiti

Marzio Della Bosca[1]

[1]m.dellabosca@campus.uniurb.it

Riassunto

Il progetto consiste in un'implementazione di Federated Learning, un approccio di apprendimento automatico che consente ai dispositivi di apprendere da dati distribuiti senza condividerli centralmente. L'architettura del sistema è basata su un modello client-server. I client eseguono l'addestramento sui propri dati locali e inviano i pesi del modello al server. Il server aggrega i pesi ricevuti dai client e coordina l'intero processo di simulazione.

1 Introduzione

Dato che il paradigma Federated Learning si presta bene in ambito ospedaliero, in cui molto probabilmente si devono gestire grandi moli di dati distribuite geograficamente (prestando molta attenzione alla privacy), il caso d'uso scelto 'e quello di una rete ospedaliera.

Ogni ospedale (client) ha il suo dataset in locale e dopo ogni sessione di addestramento manda i pesi al server il quale effettua una media ponderata dei pesi sulla base della grandezza del dataset di ogni ospedale. Dopo le medie, lato server, vengono ritornati ad ogni azienda ospedaliera la media dei pesi ponderata e il processo itera fino all'iterazione definita in principio.

Il dataset utilizzato è quello del **Diagnostic Wisconsin Breast Cancer Database**[1]. Questo dataset è composto da 569 esempi per 30 feature, il target è quello di verificare se una massa tumorale al seno è benigna o maligna.

Il progetto è stato implementato tramite python con il supporto di diverse librerie gratuite. (e.g. socket, threading, scapy, rsa, ipaddress, random, numpy, warnings, ucimlrepo.fetch_ucirepo, pandas, subprocess, platform, os, threading, time).

2 Comunicazione

Il sistema è stato progettato come un architettura client-server in cui il server rimane in ascolto e apre una connessione con tutti i client che la richiedono. La comunicazione 'e gestita attraverso lo standard 'de facto' TCP-IP e i messaggi scambiati, che vanno dai messaggi di servizio per stabilizzare la comunicazione al semplice scambio di pesi, sono criptati utilizzando una soluzione a chiave asimmetrica (RSA). Come detto prima, il progetto simula una rete ospedaliera che utilizza un modello di FL, il programma consente la simulazione sia facendo eseguire server e client sulla stessa macchina sia su macchine diverse, a patto che siano collegati alla stessa rete locale (802.11).

3 Modello di addestramento

Il modello utilizzato è stato l'SGDClassifier2, che implementa un classificatore lineare attraverso la discesa del gradiente stocastico (Stochastic Gradient Descent, SGD). Il termine "hinge" nell'ambito della funzione di perdita (loss1) indica che questo classificatore è configurato per l'addestramento utilizzando la perdita di cerniera (hinge loss), comunemente associata alle Support Vector Machines (SVM) per la classificazione binaria.

$$L = (y_i, f(x_i)) = max(0, 1 - y_i f(x_i)) \tag{1}$$

$$E(\omega, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(\omega) \tag{2}$$

Nell'equazione 2 sono descritti i dettagli matematici della procedura SDG per una classificazione binaria. Dato un set di addestramento $(x_1, y_1) \dots (x_n, y_n)$ dove $x_i \in \mathbf{R}^m$ e $y_i \in \mathbf{R}$ ($y_i \in [1, -1]$), lo scopo è quello di apprendere una funzione di score lineare $f(x) = \omega^T x + b$ con i parametri del modello $\omega \in \mathbf{R}^m$ e

intercetta $b \in \mathbf{R}$. Per effettuare previsioni per la classificazione binaria si guarda al segno della funzione di score $f(x)$. Per trovare i parametri del modello si va a minimizzare l'errore di addestramento regolarizzato. Nell'equazione R è il termine di regolarizzazione (penalità). $\alpha > 0$ è un iperparametro positivo che gestisce la forza della regolarizzazione, unico elemento di preprocessing dei dati è la normalizzazione delle feature mediante standardScaler.

4 Dataset

Il dataset, scaricato all'indirizzo `https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic` è composto da 30 feature più identificatore e target associati ad ogni esempio. Di seguito è riportata la lista degli attributi e il loro significato:

1. radius1
2. texture1
3. perimeter1
4. area1
5. smoothness1
6. compactness1
7. concavity1
8. concave_points1
9. symmetry1
10. fractal_dimension1
11. radius2
12. texture2
13. perimeter2
14. area2
15. smoothness2
16. compactness2
17. concavity2
18. concave_points2
19. symmetry2
20. fractal_dimension2
21. radius3
22. texture3
23. perimeter3
24. area3
25. smoothness3
26. compactness3
27. concavity3
28. concave_points3
29. symmetry3
30. fractal_dimension3

Si hanno circa 3 misure (1 per dimensione) per 10 valori:

- raggio: media delle distanze dal centro ai punti sul perimetro;
- texture: deviazione standard dei valori in scala di grigi;
- perimetro;
- area;
- smoothness: variazione locale nelle lunghezze del raggio;
- compattezza:

$$\frac{\text{perimetro}^2}{\text{area}} - 1.0$$

- concavità: gravità delle parti concave del contorno;
- punti concavi: numero di parti concave del contorno;
- simmetria;
- dimensione frattale: approssimazione della linea costale.

5 Testing

Data la dimensione delle immagini, screen a fini dimostrativi, queste non sono riportate all'interno della relazione ma sono all'interno della cartella di progetto.

6 Risultati sperimentali

Dato l'estimatore (molto semplice) e la piccola mole di esempi non è possibile dare nulla per certo, è stato comunque riscontrata un accuratezza migliore man mano che la simulazione proseguiva ma dato il modello e il dataset è impossibile dire se l'accuratezza ha beneficiato anche dall'operazione di aggregazione dei pesi oltre che dall'addestramento vero e proprio.

Da notare anche che, per quanto riguarda il numero di esempi a disposizione, il dataset preso in considerazione dai client era una frazione del già limitato originale. In quanto nella fase di definzione del modello veniva presa una percentuale casuale tra il 30 e l'80% in modo da avere client con numero di esempi diversi e quindi sfruttare, lato server, un ponderamento sul numero di istanze dei client.

Inoltre dato il numero così limitato di esempi ho preferito evitare la creazione di esempi sintetici (ad esempio con SMOTE).

Riferimenti bibliografici

[1] William Nick Street, William H. Wolberg e Olvi L. Mangasarian. "Nuclear feature extraction for breast tumor diagnosis". In: *Electronic imaging* (1993). URL: `https://api.semanticscholar.org/CorpusID:14922543`.