

PeerReviewFlower

Un framework per il federated learning con peer review.

Mattia Passeri

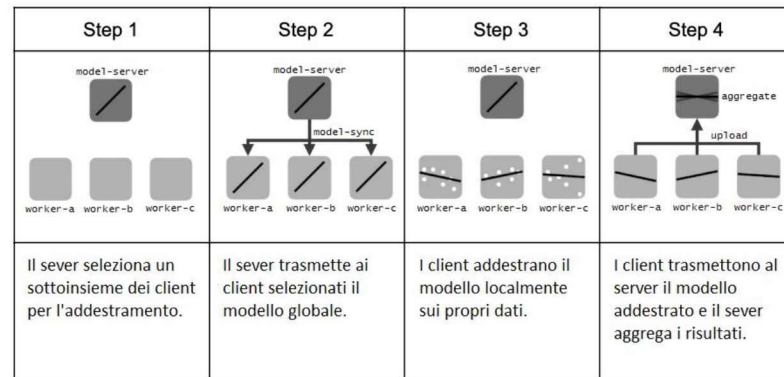
mattia.passeri2@studio.unibo.it

Introduzione

- La recente diffusione di dispositivi mobili connessi alla rete e dotati di sensori ha fatto emergere la possibilità di sfruttare le capacità di questi dispositivi di raccogliere, memorizzare e rendere disponibili grandi quantità di dati per addestrare modelli predittivi.
- Gran parte di questi dati contengono informazioni personali relative agli individui che li generano e il loro utilizzo è soggetto a restrittivi vincoli per motivi di privacy.
- Il paradigma tradizionale di gestione dei dati prevede costose attività di centralizzazione e anonimizzazione delle informazioni, ponendo così importanti limiti alle applicazioni.
- Il federated learning permette di addestrare un modello predittivo condiviso da un insieme di dispositivi distribuiti sfruttando i dati memorizzati localmente senza la necessità di raccogliarli in maniera centralizzata.
- Questo progetto riguarda lo sviluppo di una libreria software per permettere la sperimentazione di un'estensione del processo di federated learning ispirata alla procedura, comune in ambito accademico, di revisione tra pari (peer review).

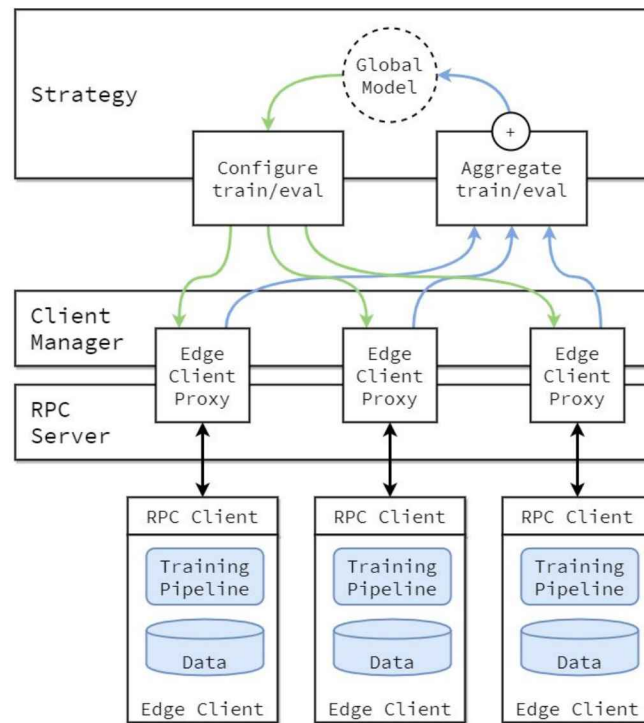
Federated Learning

- Introdotto per la prima volta da Google nel 2017.
- Un modello predittivo globale è memorizzato in un server che comunica con un insieme di client distribuiti.
- L'addestramento procede per un certo numero di iterazioni in maniera sincrona.
- Il server ad ogni iterazione seleziona un sottoinsieme dei client disponibili, i quali scaricano il modello globale, eseguono un addestramento in locale con i dati in loro possesso e al termine dell'addestramento inviano al server il proprio modello aggiornato a partire dal precedente modello globale.
- Al termine di ogni iterazione il server aggrega gli aggiornamenti provenienti dai client.
- FedAvg è il più noto algoritmo di federated learning, tipicamente applicato a modelli parametrici.



Flower

- Flower è un framework che facilita l'implementazione dei componenti fondamentali di un sistema di federated learning.
- È sviluppato in Python ed è indipendente dal framework utilizzato per eseguire gli algoritmi di machine learning.
- L'architettura di Flower separa la logica algoritmica in due parti: una eseguita sui client e una eseguita sul server.
- L'architettura di Flower realizza una versione distribuita del pattern master-worker grazie a chiamate a procedura remota dal server ai client.
- Il framework fa uso della libreria gRPC come middleware per la comunicazione e per la serializzazione dei messaggi.



Peer Review Federated Learning

- Estensione del processo di federated learning ispirata alla procedura di revisione tra pari.
- Prevede l'aggiunta di una fase di validazione degli aggiornamenti prodotti dai client durante l'addestramento.
- L'aggiornamento dei parametri del modello globale viene prodotto in due fasi che prendono il nome di fase di *training* e fase di *review*.
- La fase di training corrisponde a un round di addestramento del federated learning tradizionale, gli aggiornamenti prodotti in questa fase sono considerati provvisori.
- Durante la fase di review l'aggiornamento provvisorio prodotto da ciascun client viene sottoposto a uno o più cicli di validazione da parte degli altri client per valutarne la bontà prima che possa essere aggregato al modello globale.

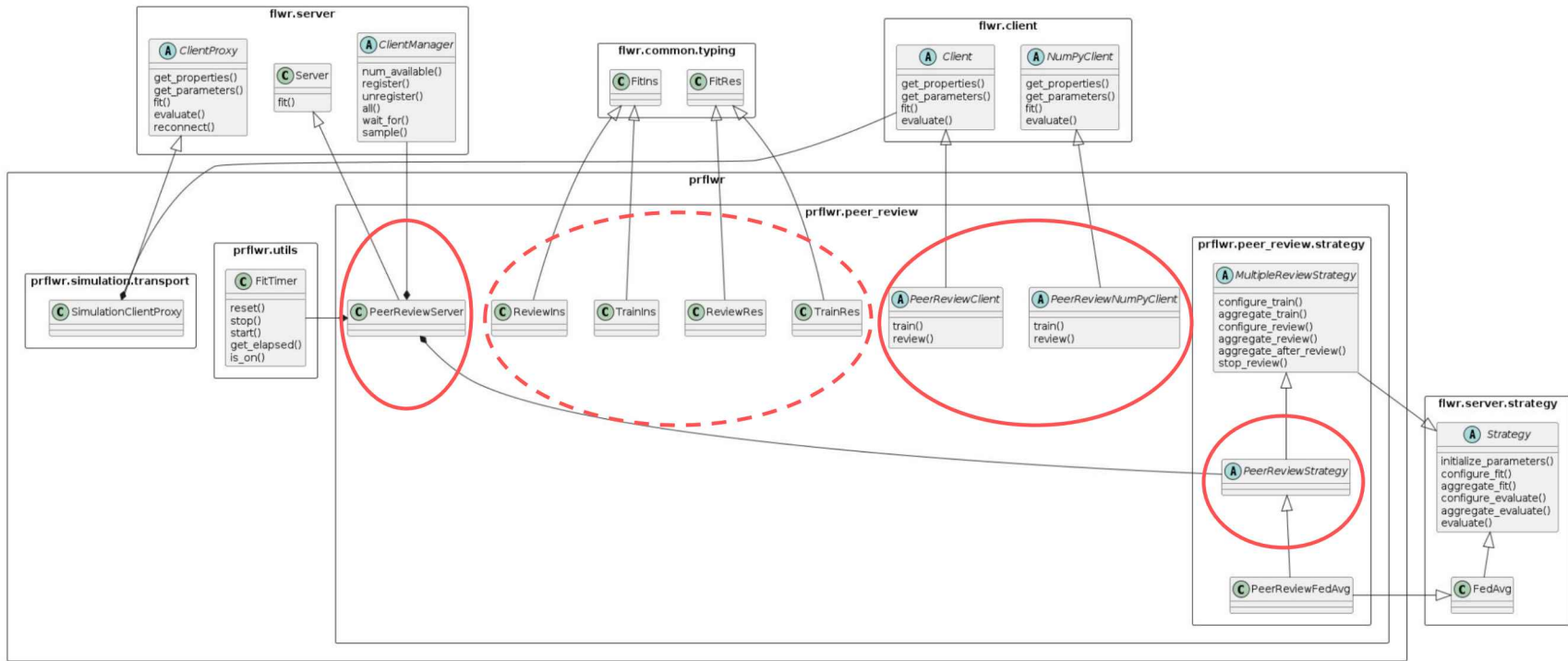
```
Input: num_rounds          # Number of FL rounds to perform
      max_review_rounds    # Max review rounds to perform each FL round

for n = 1...num_rounds:
    # training step
    select N clients
    send global model to selected clients along with training instructions
    receive model updates and aggregate them
    # review
    for m = 1...max_review_rounds:
        # review step
        select M clients
        send global model and candidate model parameters to selected clients
        receive parameters reviews and aggregate them
        if stop_review:
            break
    # candidate model parameters aggregation after multiple review steps
    aggregate candidate models parameters
```

PeerReviewFlower

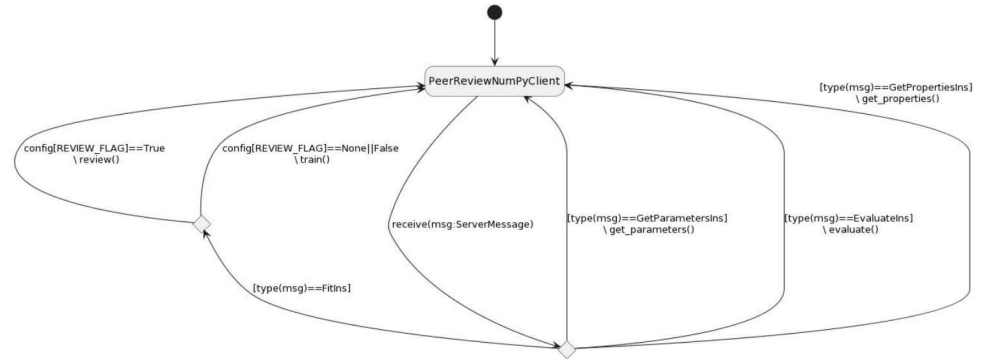
- PeerReviewFlower è la libreria sviluppata nell'arco di questo progetto come estensione di Flower, con l'obiettivo di supportare l'implementazione di algoritmi di federated learning con peer review.
- Le caratteristiche di PeerReviewFlower sono: flessibilità, semplicità e supporto al deployment sia simulato sia distribuito.
- Si è cercato di mantenere una buona compatibilità con Flower in termini di design delle interfacce in modo tale da favorire lo sviluppo successivo e l'interoperabilità.
- È prevista una logica predefinita nel server che richiama varie callback personalizzabili dall'utente, sia lato client sia lato server, che implementano una particolare strategia di addestramento.
- I componenti realizzati sono pensati soprattutto come specializzazioni dei componenti di Flower.
- Gerarchia di specializzazione tra i componenti di Flower e i componenti di PeerReviewFlower, con quest'ultimi che specializzano i corrispondenti componenti della libreria originale e risultano ad essi sostituibili se necessario.

Struttura



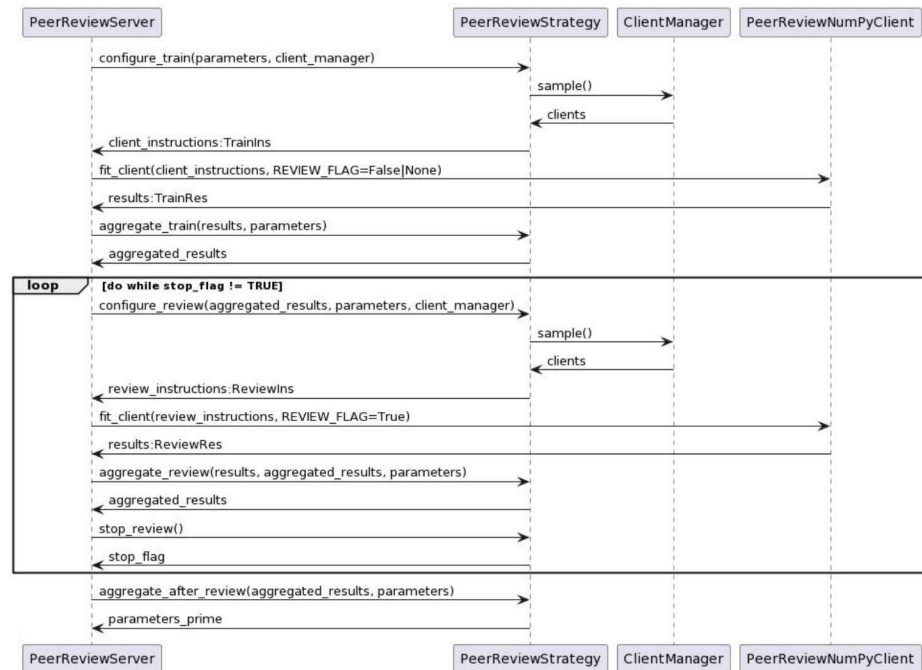
Comportamento

- Il comportamento del sistema durante un processo di addestramento federato con peer review è determinato dall'interazione tra il server e i client.
- Il server ha un comportamento proattivo, esegue il loop di addestramento alternando una fase di training, una fase di review e una fase di validazione delle performance del modello globale che può essere eseguita in maniera sia centralizzata sia federata.
- I client hanno un comportamento reattivo, attendono la ricezione di un messaggio, in base alla tipologia del messaggio ricevuto eseguono il relativo handler e rispondono al server inviando un messaggio che contiene il risultato della computazione eseguita.



Interazione

- I componenti di PeerReviewFlower sono specializzazioni dei componenti di Flower, pertanto le modalità di interazione tra essi non cambiano.
- Quello che cambia è invece la complessità dell'interazione tra entità, che aumenta per consentire l'esecuzione della fase di review.
- Per quanto riguarda i componenti di Flower che non richiedono modifiche, questi mantengono con i nuovi componenti le stesse modalità di interazione che sono previste con i rispettivi componenti della libreria originale.



Utilizzo

- Per implementare un algoritmo di peer review federated learning in *PeerReviewFlower* è necessario definire la logica di training e review lato client in una classe che estende *PeerReviewNumPyClient* o *PeerReviewClient*.
- La logica lato server dovrà essere invece definita in una classe che estende la classe astratta *PeerReviewStrategy*, implementando sia i metodi previsti dalla classe astratta *Strategy* di Flower sia quelli previsti dalla classe astratta *MultipleReviewStrategy* di *PeerReviewFlower*.
- Una volta implementata la strategia di apprendimento e la classe che modella i client, è possibile eseguire un intero processo di apprendimento federato in maniera simulata su una singola macchina oppure in maniera distribuita grazie al supporto dato da Flower.

```
class FedLSCClient(PeerReviewNumPyClient):
    def __init__(self, cid, net, trainloader, valloader):
        self.cid = cid
        self.net = net
        self.trainloader = trainloader
        self.valloader = valloader

    def get_parameters(self, config):
        return get_parameters(self.net)

    def train(self, parameters, config):
        current_round = config["server_round"]
        local_epochs = config["local_epochs"]
        set_parameters(self.net, parameters)
        optimizer = torch.optim.Adam(self.net.parameters())
        train(self.net, self.trainloader, optimizer, local_epochs)
        return get_parameters(self.net), len(self.trainloader), {}

    def review(self, parameters, config):
        current_round = config["server_round"]
        max_evaluations = config["max_evaluations"]
        gamma = config["gamma"]
        eta = config["eta0"]
        losses = []
        for i in range(max_evaluations):
            loss, num_examples, _ = self.evaluate(
                [eta * param for param in parameters], {}
            )
            losses.append(loss)
            eta = eta * gamma
        review_metrics = dict(enumerate(losses))
        return [], num_examples, review_metrics

    def evaluate(self, parameters, config):
        set_parameters(self.net, parameters)
        loss, accuracy = test(self.net, self.valloader)
        return float(loss), len(self.valloader), {}
```

FedLSBT

- Permette l'addestramento federato di modelli di apprendimento di tipo GBDT utilizzando la fase di review del federated learning con peer review.
- FedLSBT costruisce il modello predittivo come un ensemble additivo di alberi in cui il contributo di ciascun albero è pesato da una costante.
- Ciascun albero è addestrato da un solo client durante la fase di training utilizzando il dataset locale.
- Le costanti sono determinate durante la fase di review in modo tale da minimizzare una funzione obiettivo differenziabile.
- Si è realizzato un esperimento con un task di regressione e uno con un task di classificazione confrontando l'algoritmo federato con un'implementazione centralizzata di GBDT.
- Le performance misurate negli esperimenti effettuati sono risultate pari o di poco inferiori ai modelli addestrati in maniera centralizzata.

	r^2 score	MSE
Baseline	-	0.93
GradientBoostingRegressor (sklearn)	0.72	0.26
FedLSBT	0.63	0.34

Risultati ottenuti per il task di regressione.

	accuratezza	BCE
Baseline	0.50	-
GradientBoostingClassifier (sklearn)	0.81	2.47
FedLSBT	0.80	0.50

Risultati ottenuti per il task di classificazione.

Conclusioni

- Nel corso di questo progetto si è realizzato un framework che abilita la sperimentazione di un nuovo approccio al federated learning chiamato Peer Review Federated Learning.
- Insieme all'implementazione del framework si sono prodotti anche i test relativi ai singoli componenti software sviluppati e alla loro integrazione, utili a garantire la buona qualità del lavoro svolto.
- La sperimentazione del federated learning con peer review ha portato all'ideazione di un nuovo algoritmo federato di tipo gradient boosted decision trees, denominato FedLSBT, che richiede necessariamente una fase di review e, sotto determinate condizioni relative alla scelta degli iperparametri, permette di ottenere performance pari ai modelli GBDT centralizzati.
- Il lavoro svolto per questo progetto può offrire diverse nuove prospettive di sperimentazione, sia per quanto riguarda l'applicazione del peer review federated learning all'addestramento di reti neurali e altri modelli parametrici sia per quanto riguarda l'ottimizzazione e l'estensione dell'algoritmo FedLSBT.