Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 7, S. 108-112, 6, 2023

Araştırma Makalesi

IJANSER
https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 7, pp. 108-112, 6, 2023

Research Article

Federe Öğrenme Algoritmaları ve Açık Kaynak Çerçeveler

Ömer Faruk Göçgün[1*], Femin Yalcin[2] ve Aytuğ Onan[3]

[1]*Yazılım Mühendisliği / Fen Bilimleri Enstitüsü, İzmir Katip Çelebi Üniversitesi, Türkiye (ORCID: 0000-0003-1957-0794)*
[2]*Yazılım Mühendisliği / Fen Bilimleri Enstitüsü, İzmir Katip Çelebi Üniversitesi, Türkiye (ORCID: 0000-0003-0602-9392)*
[3]*Yazılım Mühendisliği / Fen Bilimleri Enstitüsü, İzmir Katip Çelebi Üniversitesi, Türkiye (ORCID: 0000-0002-9434-5880)*
[*]*(farukgocgun@gmail.com)*

(Geliş Tarihi: 04 Temmuz 2023, Kabul Tarihi: 24 Temmuz 2023)

(5th International Conference on Applied Engineering and Natural Sciences ICAENS 2023, July 10 - 12, 2023)

ATIF/REFERENCE: Göçgün, Ö.F., Yalcin, F. & Onan, A. (2023). Federe Öğrenme Algoritmaları ve Açık Kaynak Çerçeveler. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(6), 108-112.

Özet – Günümüzde dağıtık sistemler ve büyük veri, merkeziyetçi makine öğrenmesi/derin öğrenme modellerinde zaman ve donanım maliyeti gibi engellere sebebiyet vermektedir. Bu sebeple dağıtık sistemlerde çalışan yazılımların veya Nesnelerin İnterneti (IoT) cihazlarından toplanan verilerin tek bir merkezde model eğitimi veya bu verilerden sonuç elde edilmesi aynı zamanda gizlilik gibi sorunlara da sebebiyet vermektedir. Makine öğrenmesine yönelik nispeten yeni sayılabilecek olan Federe Öğrenme (Fedeated Learning), giderek küreselleşen bu dünyada veri gizliliği ve güvenliği giderek daha önemli olacaktır. Federe öğrenmede, küresel bir model oluşturmak için iş birliği yapan cihazlar ve/veya yazılımlar yinelemeli bir şekilde kendi verisini doğrudan paylaşamadan doğruluk oranını giderek arttırmaktadır. Bu ise kurumların veya firmaların büyük kaynaklar ayırarak küresel bir modeli eğitme maliyetinden kurtarmakla beraber eğitim sürecini de hızlandırmaktadır. Gizliliği koruyan veri paylaşımı özellikle sağlık, finans ve iletişim gibi sektörlerde model eğitimi için Federe öğrenmeyi öne çıkartmaktadır. Federe öğrenmenin son yıllarda araştırma odağı haline gelmiş olması sadece yeni olmasından değil, ayrıca gizliliği koruyan kanunlar, nüfus ve teknoloji kullanımındaki artış ile ideal bir çözüm olarak gelecekte kullanımının oldukça yaygınlaşacağı düşünülmektedir. Bu çalışmada, açık kaynak federe öğrenme kütüphaneleri üzerine incelemeler ile McMahan v.d.'nin FedAVG üzerinden CIFAR-10 ile yapmış olduğu çalışma Flower üzerinde simülasyonu gerçekleştirilerek karşılaştırılmalı deneysel sonuçlar sunulmuştur. Bu çalışmada yapılan deneysel sonuçlar ile, veri kümesi ve parametre ayarlarındaki değişime göre Flower çerçevesinin kullanılan algoritmanın orijinal gerçekleştirimiyle her zaman aynı doğruluk oranına ulaşmadığı görülmüştür.

Anahtar Kelimeler – Federe Öğrenme, Makine Öğrenmesi, Derin Öğrenme, Açık Kaynak Federe Öğrenme, Flower Çerçevesi

I. GİRİŞ

Geleneksel Makine Öğrenmesi (M.L.) kullanımında, tüm veriler bir veri merkezinde toplanır. Toplanan verilerle model daha sonra güçlü sunucularda merkezi olarak eğitilir. Ancak, bu veri toplama süreci genellikle mahremiyete zarar verir. Birçok kullanıcı özel verilerini şirketlerle paylaşmak istemez. Bazı durumlarda makine öğrenmesini kullanmak zordur. Gizlilik bir endişe kaynağı olmadığında bile, veri toplama zorunluluğunun makul olmadığı durumlar olabilmektedir. Örneğin, cep telefonu kullanıcıları

cihazlarıyla etkileşimde bulunarak büyük miktarda veri üretir.

Federe öğrenme, cep telefonlarının tüm eğitim verilerini cihazda tutarken ortak bir tahmin modelini iş birliği içinde öğrenmesini sağlar ve makine öğrenmesi yapma yeteneğini, verileri bulutta depolama ihtiyacından ayırır [1]. Böylece veri paylaşımı yapılmadan sadece eğitilen model parametre sunucusuna gönderilerek aynı zamanda gizlilik de sağlanmış olur. Bu çalışmada seçilen federe öğrenme algoritmaları, açık kaynak çerçeve ve kütüphanelerinin karşılaştırılması ortaya koyulmuştur.

Federe öğrenme, birbirinden farklı cihaz veya uygulamalardan kullanıcı verilerini paylaşmadan sunucu veya sunucular üzerinden sadece model ağırlıklarını alarak algoritma eğiten bir makine öğrenmesi tekniğidir.

Federe Öğrenme terimi ilk olarak McMahan vd. tarafından 2016 yılında tanımlanmıştır [2].

II. MATERYAL VE YÖNTEM

Bu çalışmada, Meta AI tarafından açık kaynak olarak geliştirilen bir derin öğrenme kütüphanesi olan PyTorch kullanılmıştır [3]. Bu sebeple CIFAR-10 veri kümesi doğrudan PyTorch veri kümesi aracılığıyla ResNet18 ağı üzerinden alınmıştır. ResNet ağı, ağların eğitimini kolaylaştırmak için, He vd. tarafından tanıtılan belirli bir sinir ağı türüdür [4].

A. Veri kümesi özellikleri

Krizhevsky vd. tarafından oluşturulan CIFAR-10 veri kümesi, sınıf başına 6000 resim olmak üzere 10 sınıfta 60000 adet 32x32 renkli resimden oluşmaktadır [5].

Bu veri kümesindeki sınıflar (uçak, otomobil, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon) tamamen birbirinden bağımsızdır. Otomobiller ve kamyonlar arasında çakışma yoktur. Örneğin, "Otomobil" sınıfı, sedanları, SUV'ları ve bu tür araçları içerir. "Kamyon" sınıfı yalnızca büyük kamyonları ve tırları içerir. Her iki sınıfta kamyonetleri içermemektedir. Amaç daha önce görülmemiş görüntüleri tanımak ve bunları 10 sınıftan birine atamaktır.

B. Metot

Bu çalışmada açık kaynak çerçevesi olarak Flower çerçevesi, simülasyon özelliği kullanılmak üzere seçilmiştir. Uygulamaların gerçekleştirilmesinde, Python programlama dili tercih edilmiş ve kodlama ortamı olarak Visual Studio Code kullanılmıştır.

Python, Guido van Rossum tarafından ilk sürümü 1991'de ortaya konan genel amaçlı bir programlama dilidir. Diğer dillere kıyasla öğrenim kolaylığı ve geniş kütüphane desteğiyle oldukça yaygın kullanıcı kitlesine ulaşmıştır [7]. PYPL programlama dilleri popülerlik indeksine göre Aralık 2019-Aralık 2020 zaman aralığında Python dilinin birinci sırada olduğu gözlemlenmiştir [8].

PyTorch kullanarak, ResNet18 ağında önceden transfer eğitimi tamamlanmış CIFAR-10 veri kümesi üzerinde konvolüsyonel sinir ağının eğitimi gerçekleştirilmiştir. Torch üzerinden veri kümesi indirilerek normalizasyon işlemleri yapılmıştır. Akabinde konvansyonel sinir ağı üzerinde model eğitimi gerçekleştirimi yapılmıştır.

Flower çerçevesinde desteklenen federe öğrenme stratejilerinden FedAvg seçilmiş ve aşağıdaki parametrelere göre çalıştırılmıştır.

Veri noktası sayısı (batch_size)	: 20
Iterasyon sayısı (Epoch)	: 5
İstemci (Total Client)	: 500
Tur (Round)	: 4000

RAY [9], AI ve Python uygulamalarını ölçeklendirmesinde kullanılan açık kaynaklı bir simülasyon çerçevesidir. RAY çerçevesi kullanan Flower üzerinde simülasyon ve eğitim gerçekleştirimi yapılmıştır.

C. Kullanılan algoritmalar

FedAvg [10], Google tarafından federe öğrenme sorunlarını çözmek için formüle edilen ilk federe öğrenme algoritmasıdır [1].

FedAvg algoritmasında istemciler, gizlilik koruması için verilerini yerel olarak tutar; istemciler arasında iletişim kurmak için merkezi bir parametre sunucusu kullanılır. Bu merkezi sunucu, parametreleri her istemciye dağıtır ve istemcilerden güncellenen parametreleri toplar [10].

Federe Öğrenme, temelde dört adımı içermektedir [11]:

1. İstemci Seçimi: Ağdaki istemciler rastgele veya istemci seçimiyle ilgili algoritmalar tarafından seçilir.

2. Parametre Yayını: Eğitilen küresel model ve parametreler seçili istemcilere gönderilir ve istemcilerdeki model, sunucunun parametreler yayınına göre güncellenir.

3. Yerel Eğitim: İstemciler paralel olarak güncellenen modele göre, yerel verileriyle yeniden eğitilir.

4. Model Toplama: İstemciler yerel model parametrelerini sunucuya geri gönderir, sunucu ve model parametreleri küresel modele doğru toplanır.

Yukarıdaki adımlar, *n* kez yinelemeli bir şekilde veya istenilen şekilde tekrarlanır [11].

D. Açık kaynak çerçeveleri

Federe Öğrenmenin 2017'de Google tarafından tanıtılmasıyla birlikte pek çok açık kaynak kodlu kütüphaneler ve çerçeveler geliştirilmiştir. Nvidia [12], OpenMined [13] topluluğu ve Intel [14] tarafından çeşitli kütüphane ve çerçeveler açık kaynak olarak geliştirilmeye başlanmıştır. Bu çalışmada Flower açık kaynak federe öğrenme kütüphanesi hızlı öğrenilmesi ve kullanım kolaylığı sebebiyle tercih edilmiştir.

NVIDIA tarafından geliştirilen Clara, sağlık hizmeti kullanım durumları için tasarlanmış bir uygulama çerçevesidir [12]. Geliştiriciler, veri bilimciler ve araştırmacılar için gerçek zamanlı, güvenli ve ölçeklenebilir birleşik öğrenme çözümleri oluşturmak için GPU kütüphaneleri, SDK'lar ve referans uygulamaları içerir [12].

OpenMined.org, ücretsiz ve açık kaynaklı bir yazılım geliştirme topluluğudur. Topluluk 2020 yılında model merkezli federe öğrenme için her biri PyGrid tarafından merkezi olarak koordine edilen çeşitli kütüphaneler geliştirmektedir [13].

OpenFL, Intel® Labs ve Pensinvanya Üniversitesi tarafından geliştirilen [14], topluluk destekli açık kaynaklı bir projedir [15]. Başlangıçta tıbbi görüntülemede kullanılmak üzere geliştirilmiş olan OpenFL, kullanım durumu, endüstri ve makine öğrenmesi çerçevesi için de kullanılacak şekilde tasarlanmıştır.

Açık kaynak kodlu Federe öğrenme çerçevesi olan Flower, mevcut makine öğrenimi iş yüklerini federe öğrenme ortamına taşımak isteyen araştırmacıları ve geliştiricileri hedeflemiştir. Flower'ın hedeflerinden biri bunu basitleştirmektir [16].

Flower ayrıca, PyTorch, TensorFlow, pandas, FastAI, PyTorch, ve scikit-learn gibi kütüphanelerine açık kaynak kodları ve makaleleri ile araştırmacılar veya girişimciler için kolaylık sağlamaktadır.

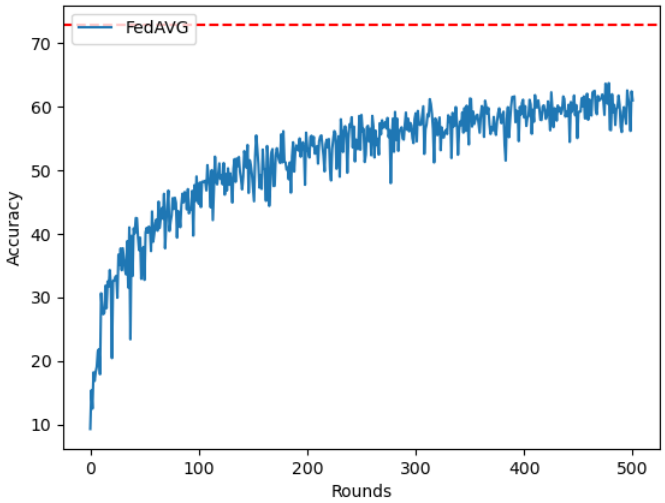
Hızlı başlangıç paketleriyle, bazı iyi bilinen federe öğrenme yayınlarından deneyleri yeniden üreten bir Flower açık kaynak kod koleksiyonu da bulunmaktadır.

III. BULGULAR

Bu çalışmada McMahan vd.'nin FedAVG üzerinden CIFAR-10 ile yapmış olduğu çalışma Flower üzerinde simülasyonu gerçekleştirilmiş ve karşılaştırılmalı deneysel sonuçlar sunulmuştur.

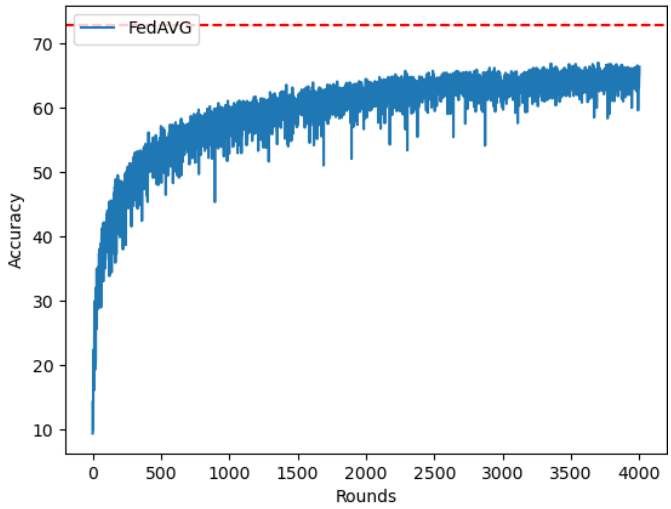
CIFAR-10 veri kümesi üzerinde, FedAvg algoritması ile veri nokta sayısı (batch size) 50 olacak şekilde 100 istemciye dağıtılarak ve 500 eğitim turu gerçekleştiren McMahan vd. %96,5 deneysel sonuç elde etmiştir. McMahan vd. aynı deneyde eğitim turunu 2000'e ve veri nokta sayısı ise 100'e çıkartıldığında doğruluk oranını %85'e düştüğünü belirtmiştir [2].

Yaptığımız çalışma sonucunda 100 istemci ve 500 iletişim turu sonucunda %62,61 sonuç elde edilmişken, tur 2000'e veri nokta sayısı 100'e çıkartıldığında ise doğruluk oranının %63,63 olduğu görülmüştür.

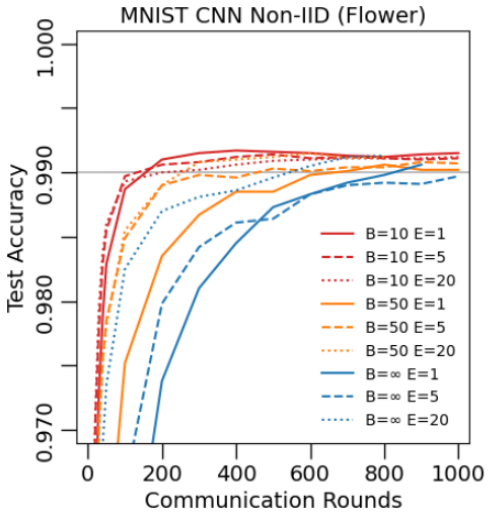


Şekil 1. 100 istemcinin 500 iletişim turu doğruluk oranları

Ayrıca, veri nokta sayısı (batch size) 20, istemci sayısı 500, eğitim turu 4000 olmak üzere her seferinde rastgele 10 istemciden veri alınarak eğitim gerçekleştirilmiştir. Eğitim tamamlandığında ise doğruluk oranlarını %66,39'a ulaştığı Şekil 2'de gösterilmiştir.



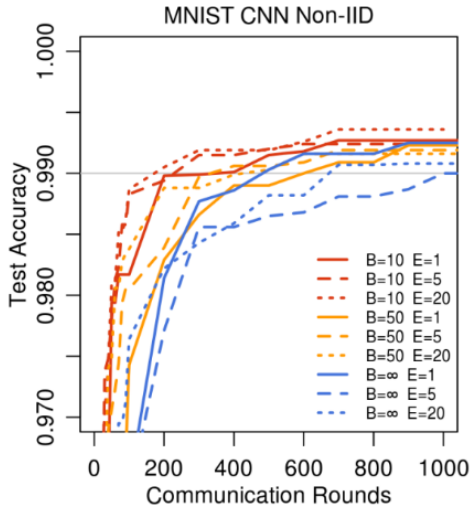
Şekil 2. 500 istemcinin 4000 iletişim turu doğruluk oranları



Şekil 4. MNIST veri kümesinde Flower deney sonuçları [17]

IV. TARTIŞMA

FedProx algoritması için Flower yazarlarından C. Beauville tarafından, MNIST veri kümesiyle yapılan çalışmada Flower'ın FedProx algoritmasının [17] asıl gerçekleştirimiyle aynı sonuçlara ulaştığı gösterilmiştir [18].



Şekil 3. MNIST veri kümesinde McMahan deney sonuçları

Bu çalışmada yapılan deneysel sonuçlar ile, veri kümesi ve parametre ayarlarındaki değişime göre Flower çerçevesinin her zaman aynı doğruluk oranına ulaşmadığı görülmüştür.

V. SONUÇLAR

McMahan vd. FedAVG algoritmasıyla ulaşmış olduğu %96,5 doğruluk oranına, Flower çerçevesinin ulaşmamış olduğu görülmüştür. Elde edilen tüm sonuçlar Tablo 1'de verilmiştir.

Tablo 1. Sonuçların karşılaştırılması tablosu

Yöntem	İstemci Sayısı	Veri nokta sayısı	Eğitim turu	Doğruluk Oranı
McMahan FedAVG	100	50	500	%96,5
McMahan FedAVG	500	20	4000	%66,39
Flower FedAVG	100	50	500	%62,61
Flower FedAVG	500	50	4000	%63,63

McMahan vd.'nin FedAVG algoritması için [18] kendi gerçekleştirimlerinde, istemci sayısı ve eğitim turundaki artışın doğruluk oranını %66,39'a düşürdüğü görülmüştür. Ancak Flower çerçevesinde ise, doğruluk oranında %1.02'lik bir artış ile %63,63'a yükseldiği görülmüştür.

Her ne kadar McMahan vd.'nin FedAvg gerçekleştirimi [18], istemci sayısı ve eğitimi turundaki artış ile doğruluk oranında düşüşe sebebiyet vermiş ise de, Flower çerçevesinin %63,63 ile ulaştığı en yüksek doğruluk

oranından daha başarımı yüksek olduğu görülmüştür.

KAYNAKLAR

[1] Federated Learning: Collaborative Machine Learning without Centralized Training Data homepage on The google's blog website [Online]. (2023) Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[2] McMahan, H.B., Moore, E., Ramage, D., Hampson, S. The PMLR Website [Online]. (2023) Available: https://proceedings.mlr.press/v54/mcmahan17a

[3] The PyTorch homepage on The Meta AI website [Online] (2023) Available: https://ai.facebook.com/tools/pytorch

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren ve Jian Sun, "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 770-778. ArXiv:abs/1512.03385

[5] The CIFAR10 dataset's website [Online]. (2020) Abailable: www.cs.toronto.edu/~kriz/cifar.html

[6] T. Sun, D. Li and B. Wang, Decentralized Federated Averaging,IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, doi: 10.1109/TPAMI.2022.3196503

[7] Malkoç B. (2012), "Temel Bilimler ve Mühendislik Eğitiminde Programlama Dili Olarak Python", XIV. Akademik Bilişim Konferansı Bildirileri, 201

[8] The PYPL PopularitY of Programming Language website [online]. (2023). Available: https://pypl.github.io/PYPL.html

[9] The RAY website's homepage [Online] (2023) https://www.ray.io

[10] Predicting Text Selections with Federated Learning homepage on The google's AI Blog website [Online]. (2023) Available: http://ai.googleblog.com/2021/11/predicting-text-selections-with.html

[11] Mammen, P. M. (2021). Federated learning: Opportunities and challenges. arXiv preprint arXiv:2101.05428.

[12] NVIDIA Clara homepage on Nvidia web site [Online]. (2023) Available: https://www.nvidia.com/en-us/clara

[13] OpenMined Syft sourcecode's readme page on The homepage of OpenMined's GitHub repository [Online] (2023) Available: https://github.com/OpenMined/PySyft

[14] G Anthony Reina, Alexey Gruzdev, Patrick Foley,vd, 2021, OpenFL: An open-source framework for Federated Learning ArXiv: abs/2105.06413

[15] Open Federated Learning (OpenFL) - An Open-Source Framework For Federated Learning repository homepage on Intel's Github website [Online] (2023) Available: https://github.com/intel/openfl

[16] Beutel, Daniel vd. "Flower: A Friendly Federated Learning Research Framework, 2020, ArXiv abs/2007.14390

[17] Sahu, A., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., & Smith, V. (2018). Federated Optimization in Heterogeneous Networks. arXiv*:* abs/1812.06127

[18] FL Starter Source Code for FedAvg MNIST homepage on the flower github repository's website [Online]. (2023) Available: https://github.com/adap/flower/tree/main/baselines/flwr_baselines/flwr_baselines/publications/fedavg_mnist